MICROCONTROLLORI Stampanti 3D Display a due digit
Visione artificiale CONVERSIONE D-A Progammazione PIC
PASCAL tastiera PS2 PIC18 GPS Raspberry MIKROBASIC

fare elettronica

www.farelettronica.com
328 Ottobre Mensile Anno XXVIII € 6,00

ESPERIMENTI DI VISIONE ARTIFICIALE

L'HARDWARE DEL RASPBERRY PI

RADIORICEVITORE FM

TIMER PER BROMOGRAFO

Tecnologia tattile resistiva e capacitiva: come funziona?

LA CONVERSIONE D/A CON I PIC18

GESTIONE DISPLAY A DUE CIFRE

PARAMETRI PER LA STAMPA 3D

Microcontrollori

USARE IL GPS CON I PIC

328 *ottobre 2012*

Zoom in

Progettare & costruire

Rispondi e... **VINCI!** *pag.* **46**

DIVERTITI E METTI ALLA PROVA LE TUE CONOSCENZE CON **ELETTRO QUIZ** E VINCI OGNI MESE ESCLUSIVI PREMI!

Pantografi CNC

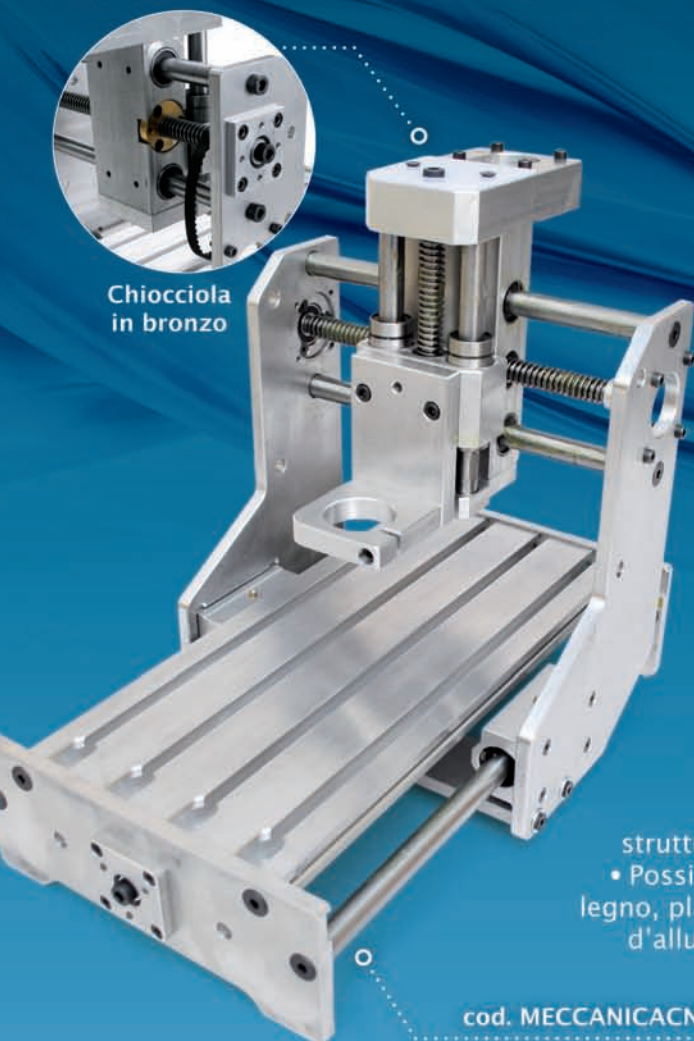
Piccoli o grandi che siano, i tuoi progetti sono sempre importanti...
... da noi trovi la soluzione più adatta per realizzarli!

Chiocciola in bronzo

Chiocciola in Delrin®

CODICE MIP 2814654

• Rigida struttura in alluminio
• Possibilità di lavorare legno, plastica, rame, leghe d'alluminio e ottone

cod. MECCANICACNC

€ 720,00

Dimensioni utili di lavoro
300 x 155 x 83 mm

cod. CNC3025

€ 1.200,00

Dimensioni utili di lavoro
300 x 250 x 100 mm

Prezzi IVA inclusa.

Schede driver 3 A

cod. 3ASSITB6560
€ 95,00

cod. 4ASSITB6560
€ 120,00

cod. PSIN30024
€ 59,00

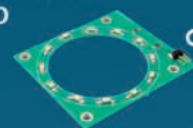
Alimentatore switching 24 Vdc / 300 W

Illuminatore a LED

cod. FT981K
€ 14,00

Motore passo-passo bipolare 2,5 A

cod. STEPMOT02
€ 44,00

Kit catene portacavo per CNC3025

cod. KITPORTACAVO
€ 114,00

Disponibile anche vasta gamma di frese e bulini

www.futurashop.it

Via Adige, 11 • 21013 Gallarate (VA)
Tel. 0331/799775 • Fax. 0331/792287

Caratteristiche tecniche di questi prodotti e acquisti on-line su *www.futurashop.it*

di MAURIZIO DEL CORSO

Energia dai segnali

È possibile recuperare energia dalle onde elettro-magnetiche di ogni genere che ci circondano? Sì, attraverso una "Rectenna"

Segnali televisivi, segnali dei telefoni cellulari, stazioni radio, Wi-Fi, Bluetooth, telecomandi... Se ci pensiamo bene, ogni giorno siamo immersi in un bagno di microonde veramente a 360 gradi. E se fosse possibile recuperare in qualche modo tutta questa energia, a prescindere dall'informazione in essa contenuta, per alimentare dei carichi?
L'argomento è di estrema attualità. Dopo le prime sperimentazioni si è riusciti a recuperare l'energia dell'ambiente mediante dispositivi detti *Rectenna*. Una Rectenna è infatti uno speciale tipo di antenna che trasforma l'energia elettromagnetica in corrente continua, utilizzabile quindi per alimentare dispositivi o per la ricarica di batterie. Nel caso in cui l'energia venga recuperata dai segnali ambientali si parla di *Energy Harvesting*, mentre se la rectenna recupera l'energia da un apposito segnale wireless si parla di vero e proprio trasferimento wireless dell'energia. L'argomento è molto affascinante e lo approfondiremo nei prossimi numeri, mostrandovi come costruire una rectenna per effettuare i vostri esperimenti.
Intanto godetevi questo nuovo numero, ricco come sempre di argomenti interessanti tra cui la stampa 3D, la visione artificiale e la programmazione dei microcontrollori.
Buona lettura!

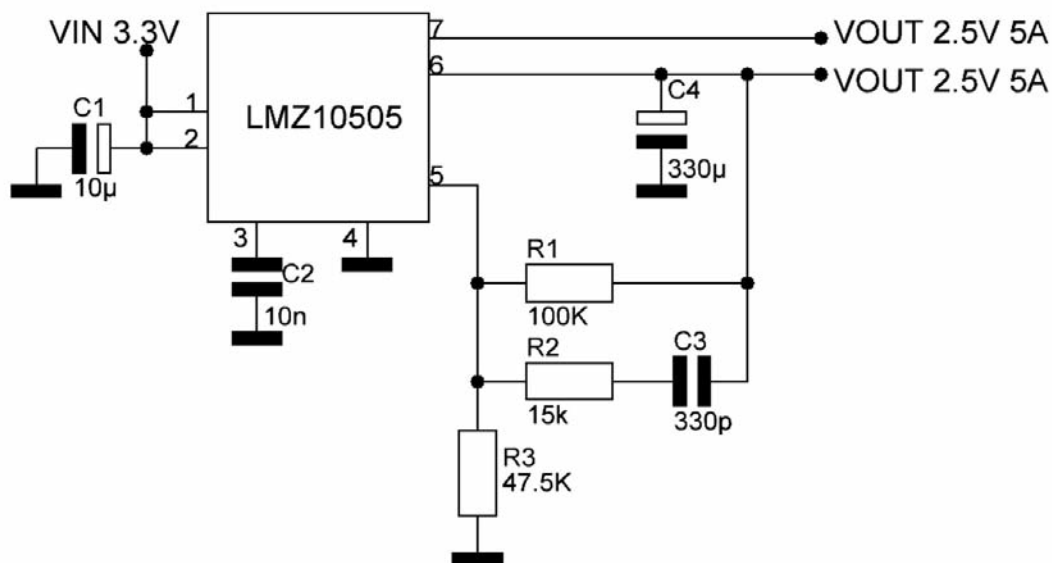
di FRANCESCO DI LORENZO

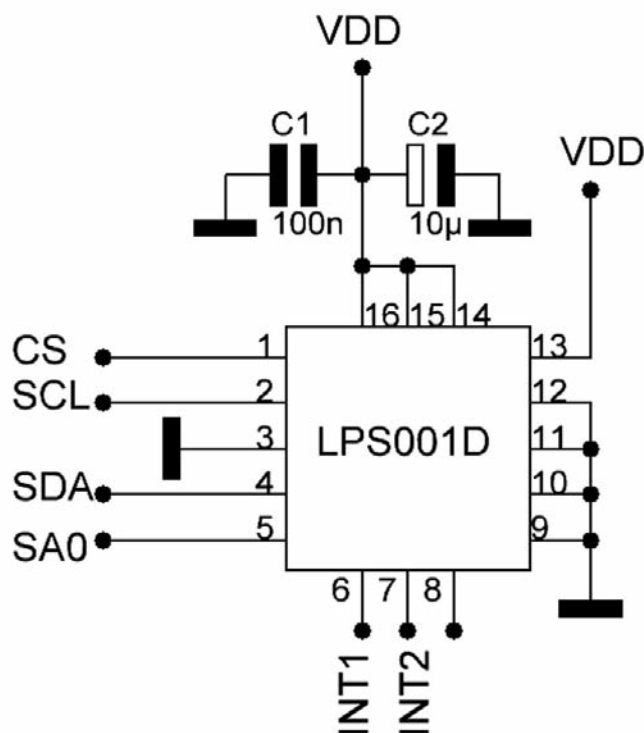
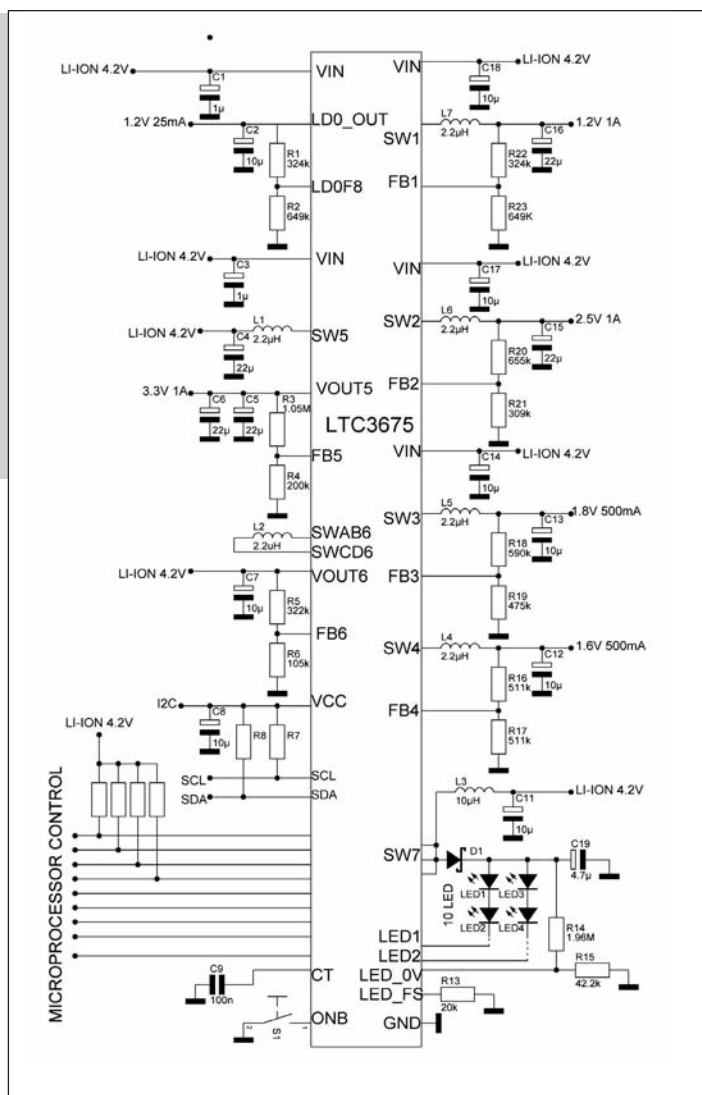
Compressore microfonico

Il LMV1099 è un dispositivo in grado di abbassare e alzare il segnale proveniente da un microfono. Per capire meglio la funzionalità basta pensare a un'autoradio che, nel momento in cui si cambia canale, abbassa leggermente il tono vocale e alza gradualmente il suono del canale selezionato. È ideale per telefoni cellulari. Può essere alimentato con una tensione variabile da 3,6 V a 5 V. Inoltre è dotata di filtri interni per eliminare i disturbi ambientali.

DC/DC Converter

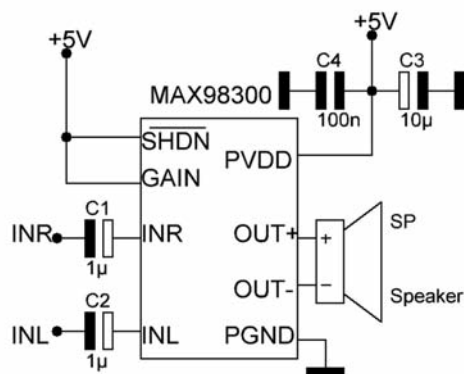
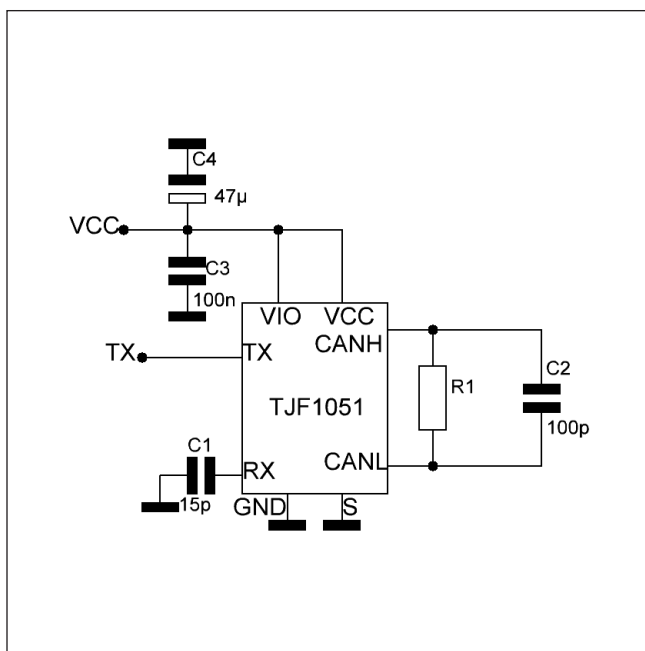
Il modulo LMZ10505 SIMPLE SWITCHER è un convertitore DC-DC in grado di erogare in uscita correnti fino a 5 A. Presenta una tensione di uscita precisa con una buona linearità. L' LMZ10505 è disponibile in un package innovativo che esalta le prestazioni termiche e permette una facile saldatura.

Il LMZ10505 può accettare un ingresso tra 2,95 V e 5,5 V e fornire in uscita 0,8 V.

SENSORE DI PRESSIONE PIEZORESISTIVO

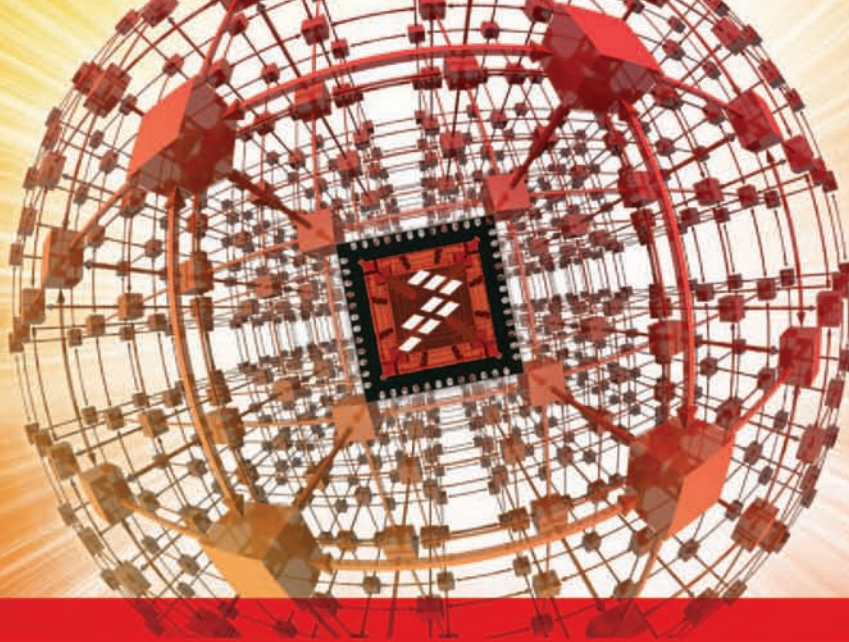
Il LPS001D è un Sensore di pressione piezoresistivo compatto. Esso comprende un elemento monolitico di rilevamento e un'interfaccia IC in grado di acquisire informazioni dalla rilevazione e di fornire un segnale digitale in uscita. L'elemento sensibile è costituito da una membrana all'interno di un substrato monosiliconico unico, prodotto utilizzando un processo dedicato e sviluppato da STMicroelectronics chiamato "VENSENS". Il processo di VENSENS permette la costruzione di una membrana monosiliconica sopra una cavità d'aria. La membrana è molto piccola rispetto alle membrane di silicio tradizionali.

convertitore DC/DC multiuscita

L'LTC3675 è un convertitore DC/DC multiuscita programmabile, a elevata efficienza, ottimizzato per applicazioni ad alta potenza. Il DC / DC è costituito da quattro convertitori buck sincroni (1 A / 1 A / 500 mA / 500 mA), tutti alimentati da un input di 2,7 V a 5,5 V. È possibile regolare 40 V con 25 mA di corrente attraverso due stringhe LED con un massimo di 10 LED ciascuno.

Amplificatore in classe D

Il MAX98300 è un amplificatore mono da 2,6 W di classe D. Questo dispositivo offre cinque impostazioni di guadagno selezionabile (0 dB, 3 dB, 6 dB, 9 dB e 12 dB). Questo integrato ha caratteristiche che riducono il numero dei componenti nelle applicazioni. Il MAX98300 è disponibile in un 8-pin TDFN-EP (2 mm x 2 mm x 0,8 mm) e un 9-bump (1,2 mm x 1,2 mm) WLP. Viene alimentato con una tensione da 3,7 V a 5 V, assorbendo pochissima corrente.

TRANSCEIVER CAN

Il TJF1051 è un transceiver CAN ad alta velocità che fornisce un'interfaccia tra un Controller Area Network (CAN) controller di protocollo e il bus fisico a due fili CAN.

Il ricetrasmettitore è stato progettato per l'alta velocità (fino a 1 Mbit/s) per applicazioni industriali, fornendo una trasmissione e ricezione differenziale per protocollo CAN. Il TJF1051 è un'evoluzione del transceiver TJA1050 CAN ad alta velocità. Esso offre migliore compatibilità elettromagnetica (EMC) e maggiore immunità alle scariche elettrostatiche (ESD).

Kinetis ARM® Cortex™-M0+ and ARM Cortex™-M4 MCUs

Scalable, ultra-low-power, mixed-signal MCUs

Our Kinetis portfolio of ARM® Cortex™ MCUs consists of multiple hardware- and software-compatible ARM Cortex™-M0+ and ARM Cortex™-M4 MCU families with exceptional low-power performance, memory scalability and feature integration. Families range from the entry-level ARM Cortex-M0+ Kinetis L series to the high-performance, feature-rich ARM Cortex-M4 Kinetis K and X series, and include a wide selection of analog, communication, HMI, connectivity and security features. All Kinetis MCUs are supported by a comprehensive Freescale and third-party hardware and software enablement system which reduces development costs and time to market.

Kinetis L Series

Kinetis L series MCUs combine the exceptional energy efficiency and ease of use of the new ARM Cortex-M0+ processor with the performance, peripheral sets, enablement and scalability of the Kinetis 32-bit MCU portfolio. The Kinetis L series frees power-critical designs from 8- and 16-bit MCU limitations by combining excellent dynamic and stop currents with superior processing performance, a broad selection of on-chip flash memory densities and extensive analog, connectivity and HMI peripheral options. Kinetis L series MCUs are also hardware and software compatible with the ARM Cortex-M4-based Kinetis K series, providing a scalable migration path to more performance, memory and feature integration.

ARM Cortex-M0+ Core

The ARM Cortex-M0+ is the world's most energy-efficient MPU and adds new energy efficiency, performance, ease of use, development and debug capabilities to its predecessor, the ARM Cortex™-M0, while maintaining 100 percent instruction set and tool compatibility.

Features

Ultra Low Power

- Next-generation 32-bit ARM Cortex-M0+ core. 2x more CoreMark/mA than the closest 8/16-bit architecture. Single-cycle fast I/O access port facilitates bit banging and software protocol emulation, maintaining an 8-bit "look and feel"
- Multiple flexible low-power modes including new compute mode which reduces dynamic power by placing peripherals in an asynchronous stop mode
- LPSCI, SPI, I²C, ADC, DAC, LP timer and DMA support low-power mode operation without waking up the core

Flash and SRAM

- Up to 32 KB flash with 64 byte flash cache, up to 4 KB RAM
- Security circuitry to prevent unauthorized access to RAM and flash contents

Performance

- ARM Cortex-M0+ core, 48 MHz core frequency over full voltage and temperature range (–40 °C to +105 °C)
- Bit manipulation engine for improved bit handling of peripheral modules
- Thumb instruction set combines high code density with 32-bit performance
- Up to 4-ch. DMA for peripheral and memory servicing with reduced CPU loading and faster system throughput
- Independent-clocked COP guards against clock skew or code runaway for fail-safe applications

Mixed Signal

- 12-bit ADC with configurable resolution, sample time and conversion speed/power. Integrated temperature sensor
- High-speed comparator with internal 6-bit DAC
- 12-bit DAC with DMA support

Timing and Control

- One 6-ch. and one 2-ch., 16-bit low-power timer PWM modules with DMA support
- 2-ch., 32-bit periodic interrupt timer provides time base for RTOS task schedule or trigger source for ADC conversion
- Low-power timer allows operation in all power modes except VLLS0
- Real-time clock with calendar

HMI

- Capacitive touch sense interface supports up to 16 external electrodes and DMA data transfer
- GPIO with pin interrupt support, DMA request capability and other pin control options

Connectivity and Communications

- I²C with DMA support, up to 100 Kb/s and compatible with SMBus V2 features
- LPUART and SPI with DMA support

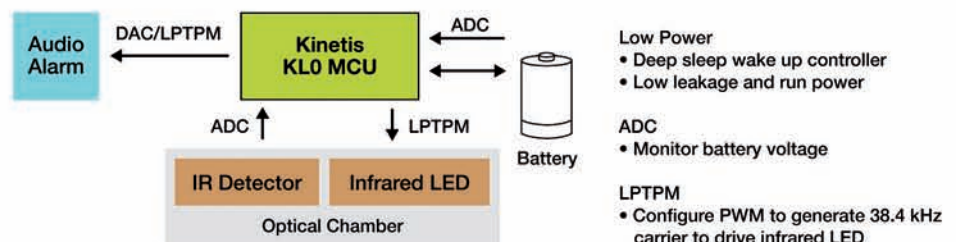
Software and Tools

- Freescale Tower System hardware development environment and low-cost demo board
- Integrated development environments
- CodeWarrior for Microcontrollers V10.x (Eclipse) IDE with Processor Expert
- IAR Embedded Workbench, Keil MDK, Atollic, CodeRed
- Runtime software and RTOS
- MQX-Lite, FreeRTOS, CodeSourcery G++ (GNU)
- Full ARM ecosystem support

Kinetis KL0 MCU Family: Smoke Detector

Low Power
- Deep sleep wake up controller
- Low leakage and run power

ADC
- Monitor battery voltage

LPTPM
- Configure PWM to generate 38.4 kHz carrier to drive infrared LED

Kinetis KL0 MCU Family

Sub-Family	Part Number	Memory			Features										√ Packages										
		CPU (MHz)	Flash (KB)	SRAM (KB)	DMA	UART	SPI	I²C	TSI	12-bit DAC	16-bit ADC w/DP Ch.	12-bit ADC	Total I/Os	Other (Key Features)	24 QFN (4 x 4, 0.5 mm)	25 WLCSP	32 LQFP (7 x 7, 0.8 mm)	32 QFN (5 x 5, 0.5 mm)	35 WLCSP	48 LQFP (7 x 7, 0.5 mm)	48 LQFN (7 x 7, 0.5 mm)	64 LQFP (10 x 10, 0.5 mm)	80 LQFP (14 x 14, 0.5 mm)	100 LQFP (14 x 14, 0.5 mm)	121 MAPBGA (8 x 8, 0.65 mm)
KL04	MKL04Z8Vxx4	48	8	1	√	√	√	√					22~28		√		√	√							
	MKL04Z16Vxx4	48	16	2	√	√	√	√					22~41		√		√	√		√					
	MKL04Z32Vxx4	48	32	4	√	√	√	√					22~41		√		√	√		√					
KL05	MKL05Z8Vxx4	48	8	1	√	√	√	√	√	√	√	√	22~28		√	*	√	√							
	MKL05Z16Vxx4	48	16	2	√	√	√	√	√	√	√	√	22~41		√	*	√	√		√					
	MKL05Z32Vxx4	48	32	4	√	√	√	√	√	√			22~41		√	*	√	√		√					

Note: Additional KL0 family members are currently under development. Visit freescale.com for the latest information.
* Proposed

For current information about Kinetis products and documentation, please visit freescale.com/Kinetis/Lseries

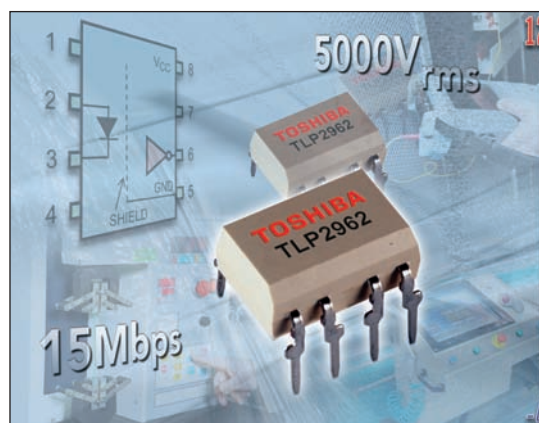
Doc Number: LSERIESKL0FS REV 0

Da Omron il nuovo sensore MEMS a infrarossi

Omron ha sviluppato un nuovo sensore termico MEMS, Micro Electro-Mechanical System, chiamato D6T, destinato a rilevare la presenza di persone senza che esse siano in movimento.
L'innovativo sensore ultra-sensibile rappresenta una straordinaria alternativa ai tradizionali sensori piroelettrici o a microonde. Il nuovo sensore D6T è un dispositivo a infrarossi che si avvale della tecnologia MEMS proprietaria di Omron. La maggior parte dei dispositivi di rilevazione presenza si basa sul movimento mentre il D6T Omron è in grado di individuare la presenza rilevando la differenza tra la temperatura di un corpo e quella dell'ambiente circostante.
Il nuovo sensore IR MEMS di Omron è disponibile in due versioni: una composta da otto celle allineate, denominata D6T-8L-06, l'altra con una matrice di celle 4x4, denominata D6T-44L-06. Grazie a questa struttura a celle multiple è possibile non solo rilevare la presenza di persone ma anche determinare la loro ubicazione. Il sensore D6T dispone di una lente in silicio che filtra i raggi infrarossi presenti nell'area di rilevamento concentrandoli sulla struttura MEMS a matrice, che rappresenta il cuore del sensore; le termocoppie collegate a ciascuna cella trasformano la differenza di temperatura generata dell'irradiamento dei raggi infrarossi in un segnale elettrico, che viene convertito in una uscita digitale I2C tramite microprocessore. Il chip MEMS è prodotto direttamente da Omron, e permette di ottenere un'elevata precisione di rilevamento della temperatura, con una risoluzione di soli 0,14 °C.

CODICE MIP 2828451

Un fotoaccoppiatore Toshiba DIP8 ad alta velocità offre 15 Mbps fino a 125 °C

Il nuovo fotoaccoppiatore compatto di Toshiba Electronics Europe soddisfa i requisiti di velocità di trasmissione, isolamento e ingombro richiesti da interfacce digitali ad alta velocità per usi industriali funzionanti sulla gamma di temperatura estesa. Il fotoaccoppiatore a canale singolo TLP2962 è disponibile in un contenitore DIP8 e gestisce velocità di trasmissione tipiche fino a 15 Mbps. Distanze garantite in aria e in superficie pari a 7 mm e 8 mm rispettivamente per il modello TLP2962 e TLP2962F garantiscono la conformità alla classe di isolamento rinforzato prevista dagli standard internazionali sulla sicurezza. La minima tensione di isolamento nominale è pari a 5000 Veff. Gli usi finali comprendono sistemi di automazione industriale, dispositivi di controllo e strumenti di misura. Il nuovo fotoaccoppiatore di Toshiba è garantito nell'intervallo termico da –40 °C a 125 °C e supporta un'ampia dinamica di tensioni di alimentazione, da 2,7 V a 5,5 V. Il dispositivo è quindi l'ideale per i nuovi sistemi a bassa tensione, nonché in sistemi esistenti realizzati con schemi di alimentazione da 5 V. Una corrente di soglia massima di soli 5,0 mA permette un funzionamento a basso consumo. In base all'applicazione, il fotoaccoppiatore TLP2962 può ridurre la corrente assorbita di un buon 75% rispetto ai dispositivi precedenti.

CODICE MIP 2828382

Vicor introduce i nuovi alimentatori ad alta densità Westcor MicroPAC

Vicor Corporation introduce i nuovi alimentatori AC-DC Westcor MicroPAC. Dotati di funzioni di correzione del fattore di potenza, i nuovi prodotti sono alloggiati in package compatti da 102x44x190 mm e sono in grado di erogare fino a 1.300 W di potenza continua con livelli di efficienza fino al 92% e densità di potenza di 1,53 W/cm3. Grazie alla possibilità di erogare tensioni di uscita semi-regolate da 12, 24, 36 e 48 VDC attraverso quattro uscite isolate configurabili in fabbrica, i nuovi Westcor MicroPAC permettono di supportare un ampio spettro di specifiche di alimentazione e offrono una soluzione ideale per le architetture distribuite destinate alle applicazioni industriali e di automazione, MIL-COTS, telecom e delle energie rinnovabili. Invece di utilizzare due alimentatori separati, i progettisti di sistema che necessitano della flessibilità necessaria per suddividere la potenza su più uscite a valori di tensione differenti possono configurare i MicroPAC sfruttando una combinazione di quattro convertitori di bus VI Chip BCM ad alta tensione ed alte prestazioni, i quali possono essere configurati con uscite singole, uscite parallele o uscite in serie con suddivisione di corrente integrata. Per le applicazioni che richiedono livelli di potenza ancora superiori, i MicroPAC possono essere configurati in array con suddivisione di corrente box-to-box.

I nuovi MicroPAC sono conformi agli standard Energy Star e offrono modalità di funzionamento eco che permettono di alimentare ogni singolo BCM solo quando necessario, assicurando un ulteriore aumento dell'efficienza energetica nell'ambito di un ampio spettro di carichi. In modalità standby, la potenza totale dissipata (TPD) è inferiore a 8 W, livello eccezionale per un alimentatore con valori nominali fino a 1.300 W.

CODICE MIP 2828325

Microchip amplia la propria linea di SRAM seriali con i dispositivi più capienti e veloci attualmente in commercio

Microchip annuncia l'ampliamento del proprio portafoglio di SRAM seriali con quattro nuovi dispositivi caratterizzati dai maggiori livelli di densità e velocità attualmente in commercio. I nuovi prodotti rappresentano anche le prime SRAM seriali con funzionamento a 5 V, tensione ampiamente utilizzata nelle applicazioni automotive e industriali. Questi dispositivi SPI low-cost da 512 Kb e 1 Mb vantano i consumi più bassi del portafoglio e sono offerti in package a 8 pin. Grazie al protocollo quad-SPI, o SQI, le nuove SRAM raggiungono velocità di 80 Mbps e operano con tempi di ciclo di scrittura prossimi allo zero, garantendo il trasferimento dati quasi istantaneo necessario per gestire funzioni grafiche, di data buffering, di data logging, di visualizzazione, matematiche, audio, video e numerose altre operazioni data-intensive.

Rispetto all'adozione di processori o microcontrollori più potenti, queste nuove SRAM da 1 Mbit consentono ai progettisti di prodotti embedded di integrare una quantità maggiore di RAM a fronte di un costo nettamente inferiore; il tutto, con consumi e pin count più contenuti in confronto alle SRAM parallele. La presenza di una SPI permette alle nuove SRAM di supportare il trend che sta privilegiando l'adozione delle interfacce seriali.

CODICE MIP 2828102

Linear Technology su Marte con Curiosity

I semiconduttori analogici ad alte prestazioni di Linear Technology Corp. sono stati selezionati dal Jet Propulsion Laboratory (JPL) della NASA per il rover Mars Science Laboratory (Curiosity). In particolare sono stati impieganti per la raccolta di grandi quantità di dati, tra cui immagini dettagliate del paesaggio di Marte e letture precise che gli scienziati utilizzano per studiare la geologia e la storia del pianeta. La NASA ha dichiarato che lo scopo della missione è determinare se sul pianeta rosso vi siano condizioni adatte a sostenere la vita e cercare eventuali forme di vita passata. I dispositivi di Linear Technology sono stati scelti per la missione su Marte per le prestazioni, la precisione e l'affidabilità, oltre che per la loro capacità di resistere in ambienti difficili, sia nella fase di volo che sulla superficie di Marte. I prodotti di Linear Technology sono installati sia su Curiosity che sulla navicella usata per il trasporto del rover: regolatori di commutazione per alimentare la strumentazione di bordo, convertitori analogici-digitali per il controllo delle telecamere che consentono al rover di "vedere" il paesaggio e digitalizzare le immagini da inviare sulla Terra e amplificatori operazionali per amplificare i segnali a livelli precisi e inviare dati accurati sulla composizione del pianeta rosso. Oltre che su Curiosity i prodotti di Linear Technology erano presenti anche sui rover Spirit e Opportunity, atterrati su Marte nel 2004, e sulle sonde Mars Global Surveyor, Mars Pathfinder, Cassini, Deep Space 1 e Mars Odyssey. Linear Technology fornisce alla NASA/JPL circuiti integrati analogici che offrono il massimo in fatto di prestazioni, precisione e affidabilità in package ridottissimi. I prodotti forniti da Linear sono resistenti alle radiazioni.

CODICE MIP 2827775

MOBOTIX lancia sul mercato la nuova telecamera S14 FlexMount, la prima telecamera emisferica flessibile

Si tratta di una novità assoluta a livello mondiale. La nuova MOBOTIX S14 FlexMount è infatti la prima telecamera emisferica flessibile al mondo, disponibile sia nella versione Mono (S14M), sia in quella Dual (S14D). Nella versione Dual, la S14 è dotata di due distinti obiettivi con ripresa emisferica e microfono integrato, connessi all'alloggiamento principale tramite un semplice cavo di collegamento. Ciò consente di coprire completamente con una sola telecamera S14 due zone adiacenti o limitrofe. Grazie alle dimensioni ridotte, e ai due obiettivi disponibili sia in bianco sia in nero, la telecamera può essere installata in maniera discreta in qualsiasi tipologia di contesto ed è stata studiata appositamente per applicazioni in ambito domestico. Una sola S14D può sorvegliare contemporaneamente due ambienti distinti (anche nel caso in cui si tratti di un ambiente interno e di uno esterno) e i due obiettivi separati sono in grado di generare immagini panoramiche a 180°, ciascuna con risoluzione da 3,1 megapixel, con correzione automatica della distorsione. Un'altra grossa novità è che la S14 è utilizzabile come prima telecamera emisferica giorno-notte. Quando entrambi i moduli (sensori bianco-nero e a colori) vengono montati uno accanto all'altro in modo da coprire la stessa area, la telecamera sceglie automaticamente la modalità ideale in base alla situazione della luminosità. Ciò consente di ottenere un'ottima fedeltà dei colori in presenza di luce diurna e un'alta sensibilità in ambienti bui. Il software completo per la configurazione e l'utilizzo della telecamera è direttamente integrato nella telecamera. lnoltre, il software di gestione video può essere scaricato gratuitamente dal sito Web di MOBOTIX.

CODICE MIP 2828471

ANALISI, VERIFICA E COLLAUDO DEGLI IMPIANTI FOTOVOLTAICI CON PROFITEST PV DI GOSSEN METRAWATT

In armonia con il motto: "Collega – alimenta – misura – leggi – fatto", Gossen Metrawatt presenta al mercato della generazione fotovoltaica il PROFITEST PV, nuovo strumento di misura per l'analisi, la verifica e il collaudo degli impianti fotovoltaici.

Il PROFITEST PV è uno strumento ideale per la misura della curva caratteristica I-V e della potenza di picco sia dei singoli moduli, sia delle stringhe, con impostazione automatica dei campi di misura fino a 1000 V / 20 A. Lo strumento accerta direttamente on-site la potenza di picco, la resistenza di serie e parallelo con una sola misura e senza l'introduzione di alcun dato concernente il modulo per la determinazione dell'efficienza dei pannelli fotovoltaici. Alcune caratteristiche del PROFITEST PV: rilevazione accurata della curva caratteristica I-U di singoli moduli e stringhe mediante il metodo di misura capacitivo; misura della corrente di cortocircuito ISC, della tensione a vuoto VOC, potenza di picco istantanea della cella solare PMAX, resistenza serie RS e resistenza parallelo interna RP; processo di elaborazione brevettato per la valutazione dell'efficienza dei moduli e per la determinazione delle caratteristiche interne a partire solo dalla curva I-U; valori visualizzati: potenza di Picco PPK, resistenza interna serie RS e parallela RP di moduli e stringhe; valori istantanei: UPmax, IPmax, Pmax, UOC, ISC, FF, TMOD, TREF, ETRMS, diagramma curva caratteristica I-U; conversione automatica dei valori rilevati a quelli di riferimento standard (STC; Misura di tensioni generate fino a 1.000 V DC, correnti fino a 20 A DC, potenza fino a 20 kW; memoria interna per l'archiviazione di parecchie migliaia di valori.

CODICE MIP 2828469

AVX presenta una nuova serie di condensatori DC a film di collegamento

AVX Corporation, produttore leader di avanzati componenti passivi e soluzioni di interconnessione,
ha introdotto una nuova linea di condensatori dielettrici DC a film di collegamento in polipropilene. Denominati serie FB, tali condensatori a 2 conduttori presentano ratings di tensioni da 550 V a 1.200 V, una gamma delle capacitanze da 0,68 µF a 75µF, hanno conformità RoHS, e sono disponibili in una gamma di dimensioni della capsula contenente oltre 100 nuovi ratings. Le applicazioni idonee alla serie FB includono gli alimentatori di potenza DC e gli invertitori per l'energia solare, i comandi elettrici e i sistemi industriali di alimentazione di potenza.

Classificati per l'impiego in presenza di temperature da –40 °C a +100 °C, i condensatori DC a film di collegamento serie FB di AVX sono disponibili in 14 dimensioni di capsula con tre opzioni standard del passo del conduttore: 27,5 mm, 37,5 mm e 52,5 mm.

CODICE MIP 2828108

I rivoluzionari interruttori capacitivi Molex trovano impiego in lavatrici di fascia alta

Molex Incorporated ha formato una partnership con un grande produttore internazionale di elettrodomestici per lo sviluppo di pannelli tattili capacitivi destinati a due modelli di lavatrici di fascia alta della classe di risparmio energetico A. Gli interruttori e i pannelli capacitivi di Molex utilizzano una tecnologia di rilevamento di campo capacitivo che permette di realizzare interfacce utente prive di componenti mobili per una serie di applicazioni in prodotti medicali, industriali, consumer e per elettrodomestici.
I progetti dei pannelli con interruttori capacitivi integrano i circuiti a stato solido in una robusta interfaccia altamente resistente a molteplici pressioni dei tasti, all'acqua, allo sporco, ai contaminanti e alle IEM. Gli overlay e i pannelli capacitivi Molex consentono di pulire rapidamente gli elettrodomestici e sono disponibili con una serie di opzioni per scopi estetici fra cui decorazioni stampate (IMD, In-Mould Decoration) ed elementi in vetro o quasi in qualunque materiale non conduttivo. Entrambi i modelli di lavatrice sono dotati di più cicli di lavaggio, che richiedono l'integrazione di numerosi sensori e icone capacitivi essenziali in spazi limitati. Un unico microcontroller a 44 piedini gestisce tutte le funzioni del pannello. Un sistema di comunicazione mediante ricevitore/trasmettitore asincrono universale per le segnalazioni dei sensori e il controllo dell'illuminazione garantisce livelli costanti di prontezza di risposta e, nelle modalità con luminosità e attenuazione controllate dall'utente, di illuminazione del pannello. Il tasto di accensione è stato integrato nei pannelli capacitivi Molex in modo da offrire ulteriori risparmi energetici quando l'elettrodomestico non è in uso; fino a quando non si aziona il tasto di accensione, le icone rimangono infatti nascoste dietro un frontalino dall'aspetto inerte.

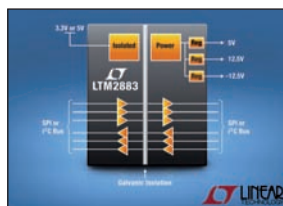
CODICE MIP 2827561

Peregrine Semiconductor svela gli Switch RF SPDT

Peregrine Semiconductor Corporation, fornitore fabless di circuiti integrati (IC) di elevate prestazioni per radio frequenza (RF), ha annunciato la disponibilità dello switch RF SPDT PE42421, nel suo package in miniatura 6-lead SC-70. Con le piccole dimensioni del suo package, l'eccellente prestazione ESD di 2 kV sulla porta RFC, supporto di basse tensioni fino a 1,8 V VDD, e basso consumo di potenza pari a 9 microampère (tip.), il nuovo switch è perfetto per applicazioni portatili alimentate a batteria, per telematica automotive, come telecomandi per veicoli e chiamate di emergenza.

Lo switch reflective UltraCMOS 50 Ohm si contraddistingue per una elevata linearità di 50 dBm IIP3 e logica di controllo CMOS on-chip, low-voltage. Può essere controllato utilizzando input sia single-pin sia complementary-control. Inoltre, lo switch offre una bassa perdita di inserzione di 0,35 dB @ 1,0 GHz e 0,50 dB @ 2,0 GHz, con un elevato isolamento pari a 30 dB @ 1 GHz e 20 dB @ 2 GHz. Utilizzando una tensione di alimentazione nominale di +3 V, un input tipico di 1 dB può essere raggiunto un punto di compressione di +33,5 dBm.

CODICE MIP 2828473

Isolatore SPI/digitale o µModule I2C fornisce tre rail di potenza isolati

Linear Technology Corporation presenta l'LTM2883, un isolatore SPI/digitale a 6 canali o µModule I2C digitale con potenza regolata su tre rail per i sistemi da 3,3 V e da 5 V. Nelle applicazioni dei sistemi industriali, i potenziali di massa possono variare notevolmente, spesso superando il range tollerabile, interrompendo le comunicazioni o addirittura distruggendo i componenti. L'LTM2883 interrompe i loop di massa separando elettricamente i segnali di comunicazione, isolando l'interfaccia a livello logico su ciascun lato di una barriera di isolamento induttivo interno che supporta un range di tensioni common mode molto ampio, fino a 2.500 VRMS. Il convertitore DC-DC isolato a basse emissioni elettromagnetiche dell'LTM2883 alimenta l'interfaccia di comunicazione e fornisce uscite di alimentazione regolabili di 5 V, +12,5 V e –12,5 V per alimentare i convertitori di dati nei sistemi di acquisizione dati. Con un isolamento galvanico di 2.500VRMS, alimentazione secondaria onboard e un'interfaccia di comunicazione che funziona fino a 20 Mbps, l'LTM2883 non richiede componenti esterni e fornisce una semplice soluzione µModule per le comunicazioni dati isolate. L'LTM2883 è disponibile con due diverse versioni dell'interfaccia di comunicazione. L'LTM2883-I è conforme alla specifica I2C fino a 400 kHz con dati seriali bidirezionali (SDA) più un orologio (SCL) e tre segnali logici CMOS isolati supplementari che funzionano fino a 20 Mbps. L'LTM2883-S è conforme a SPI e offre in totale sei canali di comunicazione dell'isolatore CMOS digitale. Tutti i canali funzionano fino a 20 Mbps e includono tre segnali diretti (/CS, SCK e SDI) e tre segnali inversi (SDO, DO1 e DO2). Se configurati per le comunicazioni SPI, la velocità di clock massima è 8 MHz per la comunicazione unidirezionale o 4 MHz per il funzionamento bidirezionale circolare.

CODICE MIP 2828477

MikroElektronika ha rilasciato una nuova serie di schede accessorie che utilizzano l'innovativo socket MikroBUS.

Il socket MikroBus è stato introdotto sin dal rilascio dalla scheda EasyPIC7 con lo scopo di consentire una facile integrazione. Il nuovo socket è stato creato prendendo il set di pin più utili allo sviluppo ed è stato prodotto un pinout standard che comprende: i pin per l'alimentazione, SPI, I2C, analogico, UART, Interrupt, PWM e reset.
Le nuove schede accessorie prodotte (denominate schede "Click") sono:
- **USB SPI Click: aggiunge una comunicazione seriale SPI tramite il cavo USB;**
- **THERMO click: è dotato di convertitore MAX31855K termocoppia-to-digital e connettore PCC-SMP per sonde a termocoppia di tipo K.;**
- **FLASH click: una soluzione semplice e compatta per aggiungere la memoria Flash al proprio dispositivo;**
- **EEPROM click: è dotato di una EEPROM in packaging DIP con tecnologia CMOS seriale a 8 K altamente affidabile e dalle alte prestazioni;**
- **7seg click: semplice soluzione per aggiungere un display a 2 cifre a sette segmenti sul proprio dispositivo;**
- **USB UART click: aggiunge una comunicazione seriale UART tramite cavo USB.**

CODICE MIP 2828475

GMM 886: grifo Mini Modulo Microchip PIC16F886

Potente ed economico, il nuovo Mini Modulo di grifo è basato sul controllore Microchip PIC16F886 in un contenitore DIP da 28 piedini e, con 14,3 K di Flash, è un completo SBC programmabile In Circuit con linguaggi evoluti come C, BASIC, ecc. Il GMM 886 è in grado di essere utilizzato, come Macro Componente, direttamente sulla scheda dell'utente. È alimentato a 5 Vdc e a bordo ha tutto ciò che serve per funzionare e per comunicare tramite una linea seriale, una a livello TTL oppure in RS232.
L'enorme vantaggio di questo Mini Modulo è possedere un programma di BootLoader che gli consente di essere programmato / cancellato usando la sola linea di comunicazione seriale. Questo significa che qualsiasi sperimentatore può farsi un circuito, perfettamente funzionante, e programmarlo usando la sola linea seriale.
È il componente ideale per risolvere i problemi di automazione sia industriale sia domestica. A questo scopo è sufficiente provvedere, tramite una circuiteria esterna, a bufferizzare le linee di I/O disponibili. Se non si vuole costruire hardware è possibile utilizzare il GMB HR84 il quale provvede ad alimentare e bufferizzare le linee di I/O del GMM 886 con 8 ingressi optoisolati, e visualizzati tramite LED, indifferentemente usati come ingressi NPN o PNP, 4 relè da 5 A, una linea di seriali TTL o RS232, una linea in I²C BUS. Se, invece, servono più risorse analogiche si può ricorrere alla GAB H844 che con le sue 8 linee di A/D converter, 4 opto-in e 4 relè output diventa una risorsa ideale.
Molto interessanti anche le dotazione di software di programmazione, e di esempi, che comprendono vari Compilatori C, BASIC tra cui il PIC BASIC Pro Compiler, MicroCode Studio PRO ecc.
Il GMM 886 è distribuito da Grifo - Via dell'Artigiano 8/6 - 40016 San Giorgio di Piano (BO).

CODICE MIP 2828402

EXPO RADIO POTENZA

L'Expo Radio & Informatica è la mostra mercato dedicata all'elettronica e all'informatica, compagne ormai indispensabile nel lavoro e negli hobby. Ideata per consentire agli operatori del settore una migliore valutazione dell'offerta delle nuove tecnologie, essa offre anche a un pubblico più eterogeneo, in cerca di buone occasioni commerciali, tutta l'esperienza e la professionalità acquisita nel campo delle telecomunicazioni e dell'informatica. In contemporanea è prevista una mostra di Protezione Civile per sottolineare l'importanza delle comunicazioni radio in eventi calamitosi, indispensabili nella gestione dell'emergenza.

Saranno presenti espositori provenienti dall'intero territorio nazionale con: radioricetrasmettitori per CB – OM, computer, telefonia, antenne e parabole per TV sat, Hi-Fi car, strumenti di misura e utensili vari, surplus, radio d'epoca, editoria specializzata.

Dove: Potenza, area industriale di Tito Scalo
Quando: 12-14 Ottobre 2012
Orari: 9.00/13.00-16.30/20.30
Organizzazione: ENTE FIERA AUTONOMO BASILICATA
Info: www.basilicatafiere.it

CODICE MIP 2810469

Makers Italy 2012

Grande evento italiano dedicato al mondo dei *makers*, dei creativi digitali e tecnologici, dei *garage innovators*, ma anche a tutti gli appassionati del *Diy* (do-it-yourself) che, con le loro innovazioni e i loro nuovi modelli di business, stanno dando vita a quella che molti commentatori non esitano a definire una nuova "rivoluzione industriale". Una rivoluzione che vede protagonista la cultura digitale, la quale, dopo aver sovvertito il mondo dei bit e quindi l'editoria, la musica e i video attraverso Internet, secondo Chris Anderson, direttore di Wired Usa: « ora sta per trasformare il mondo degli atomi, quindi degli oggetti fisici », offrendo infinite opportunità imprenditoriali a giovani creativi e brillanti, che saranno impegnati a generare business partendo dall'open source e dalla collaborazione tra persone.

Dove: Fiera Milano Rho (MI)
Quando: 9-11 novembre 2012
Orari: dalle 9.00 alle 18.00
Organizzazione: Italia Crea
info: www.mkersitaly.it

CODICE MIP 2826019

Automazione in fiera

SAVE è un appuntamento innovativo che unisce una parte espositiva in fiera a una forte componente formativa. Area espositiva dove incontrare agli stand i principali leader di settore, centinaia di convegni e workshop accessibili per gli operatori qualificati, cinque eventi internazionali e un evento speciale in contemporanea, gli operatori professionali accedono gratuitamente alla mostra e usufruiscono di tutti i servizi. L'esigenza percepita oggi è rendere adeguate le modalità fieristiche, renderle attuali e sempre più proficue per creare una fiera sull'automazione e sulla strumentazione utile agli operatori.

Dove: Verona
Quando: 24-25 ottobre 2012
Orari: dalle 9.00 alle 18.00
Organizzazione: EIOM
info: www.exposave.com

CODICE MIP 2823210

AUTOMAZIONE IN FIERA

Electronic Days, Fiera dell'Elettronica in genere, Mostra Mercato con vendita consentita.

Dove: Mantova, presso PalaBam, Via M.Gioia Zona Favorita
Quando: 17-28 Ottobre 2012
Orari: -
Organizzazione: ITALFIERE srl
info: www.italfiere.net

CODICE MIP 2810473

Fiera dell'informatica elettronica e radiantis

Grandi occasioni per tutti, a partire dai supporti per registrazioni audio/video per finire con l'informatica, in continua crescita, e il radiantismo. Uno spazio considerevole, con più di trenta espositori, è stato riservato all'hobbistica e all'elettronica con la mostra scambio del radioamatore. Anche gli strumenti di misura e le apparecchiature elettroniche sono ben rappresentati dalla presenza di molti espositori specializzati.

Dove: Rovigo
Quando: 27-28 Ottobre 2012
Orari: Sabato 9:00/19:00
Organizzazione: AREA REBUS
info: www.arearebus.com/fiera

CODICE MIP 2810495

di DAVID ASKEW

Tecnologia TATTILE RESISTIVA *e* CAPACITIVA

Il boom del mercato dei sensori tattili ha dato vita a una vasta gamma di tecnologie tattili, tra cui i touch-screen resistivi realizzati in differenti design di ottimo rapporto qualità/prezzo, i metodi capacitivi reciproci proiettati, utilizzati negli smartphone, e i dispositivi a base piezoelettrica impiegati per la creazione di touch-display di grandissime dimensioni. Ma come funzionano e perché vengono utilizzati? Esaminiamo due delle più comuni tecnologie touch: sensori tattili resistivi e sensori tattili capacitivi.

TOUCH-SCREEN RESISTIVI

Sebbene il mercato tattile capacitivo possa vantare i tassi di crescita più elevati, i touch-screen resistivi sono ampiamente disponibili e rimangono di uso comune. La tecnologia tattile resistiva è relativamente semplice ed è in grado di garantire una risoluzione più elevata a costi inferiori rispetto ai touch-screen capacitivi.
I sensori tattili resistivi si distinguono dagli altri metodi tattili in quanto il sensore è attivato meccanicamente. In altri termini, vi sono due strati conduttivi separati da un'intercapedine d'aria ed è necessario esercitare una forza di compressione per piegare lo strato superiore facendolo entrare in contatto con quello inferiore. Questo può rappresentare un problema o un vantaggio, a seconda dell'applicazione. Il sensore tattile è semplice da interfacciare e può essere attivato mediante un'ampia gamma di oggetti, quali uno stilo, una gomma per matita, l'unghia di un dito o perfino un dito guantato.
Tuttavia, la necessità di uno strato superiore flessibile comporta il fatto che lo schermo sia soggetto a usura meccanica e, sebbene non danneggiabile da polvere o acqua, sia più suscettibile a graffi e danni dovuti a oggetti appuntiti. I tipici touch-screen resistivi presentano, inoltre, il livello più basso di qualità ottica, trasmettendo solo circa l'80% della luce del display sottostante. Eppure il costo abbordabile, la semplicità e la versatilità dei touch-screen resistivi spesso fanno di questa tecnologia un'opzione allettante per varie esigenze di design.

Come funzionano?

I due strati conduttivi di un touch-screen resistivo devono essere trasparenti e pertanto sono in genere composti da ossido di indio-stagno (ITO). Lo strato inferiore è depositato su un substrato di vetro ed è separato dallo strato superiore in ossido di indio-stagno da puntini distanziatori trasparenti. La superficie dello schermo è rivestita da uno strato flessibile di isolamento protettivo che spesso presenta proprietà di resistenza ai graffi e anti-abbagliamento.
Esistono vari metodi per determinare la po-

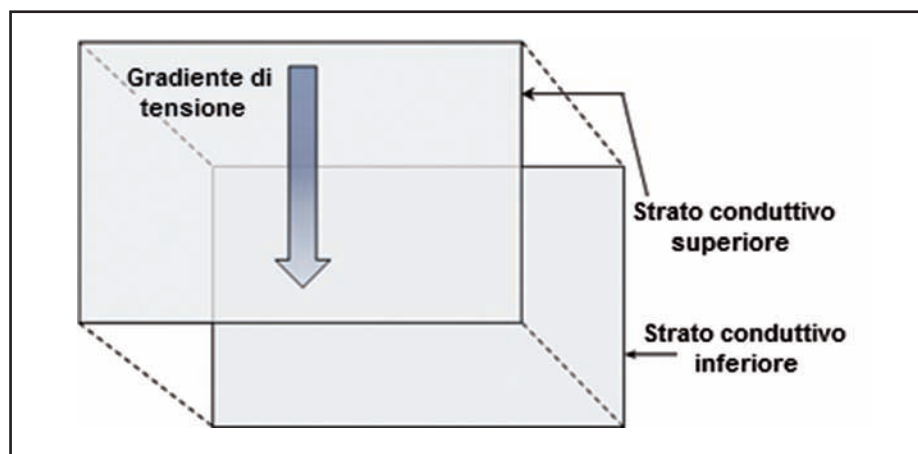
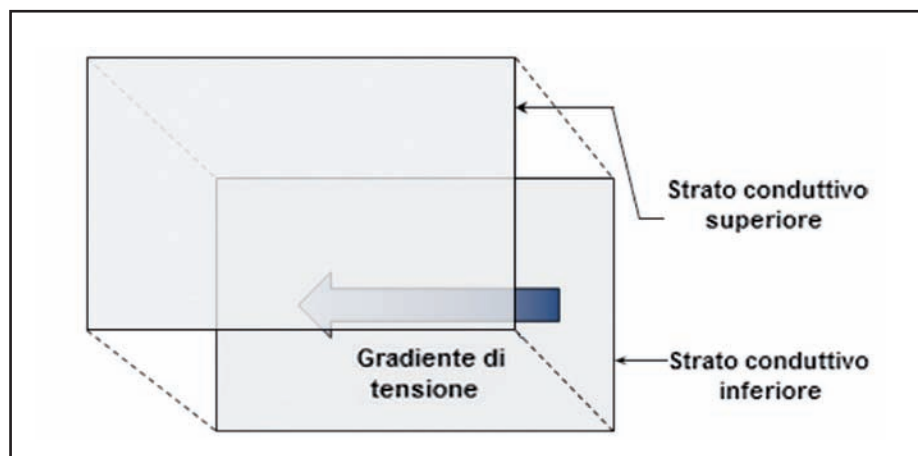



Figure 1 e 2: touch-screen resistivo a 4 fili.

sizione tattile di uno schermo resistivo. Il metodo più semplice utilizza solo quattro fili: due collegati ai lati sinistro e destro del conduttore inferiore e due collegati al lato superiore e inferiore dell'altro conduttore. Un gradiente di tensione viene applicato attraverso uno degli strati ITO uniformemente resistivi (ad esempio lo strato inferiore) e quando viene effettuato il contatto con lo strato superiore il circuito risultante assomiglia a un divisore di tensione. La tensione viene rilevata in corrispondenza dello strato superiore per determinare la coordinata X della posizione tattile. Questo processo viene ripetuto per l'asse Y applicando un potenziale attraverso il conduttore superiore e misurando la tensione dello strato inferiore.

I touch-screen resistivi a 4 fili e i corrispondenti controller sono onnipresenti e a basso costo. Tuttavia, poiché questa tecnologia utilizza lo strato flessibile per la misurazione, l'usura sullo strato superiore in ITO può modificare le caratteristiche di resistenza del conduttore, causando una riduzione potenzialmente significativa della precisione. Negli schermi di grandi dimensioni questo inconveniente e il cosiddetto "drift" del sensore sono maggiormente evidenti. Il metodo a 4 fili è pertanto preferibile per i touch-screen di piccole dimensioni **(figure 1 e 2)**.
I touch-screen resistivi a 5 fili sono un tentativo di risolvere il problema dell'utilizzo di uno strato superiore flessibile per le misurazioni. Questa tecnologia utilizza

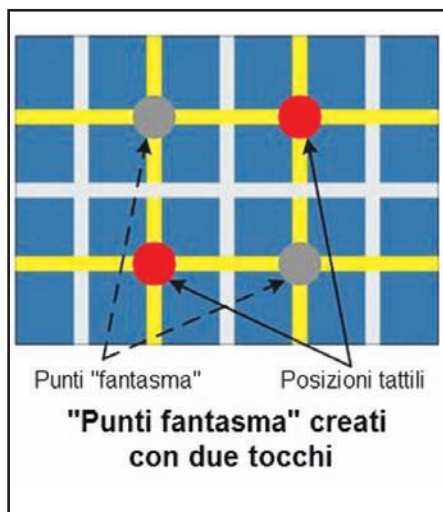



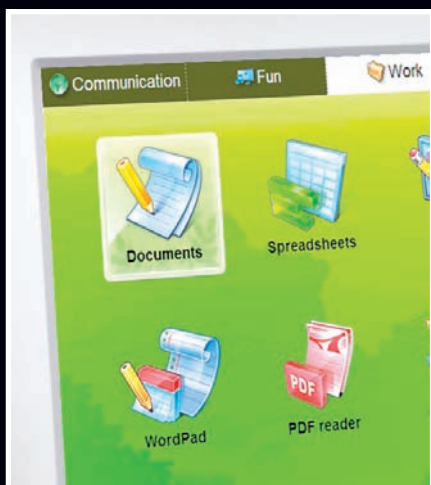
Figura 3: schermo tattile auto-capacitivo.

sia i fili "sinistro-destro" che "superiore-inferiore" dello schema a 4 fili, sia elettrodi collegati invece ai quattro angoli dello strato inferiore stabile, con il quinto filo che funziona come sensore di tensione dello strato superiore. Lo strato inferiore viene pertanto utilizzato per le misurazioni sia dell'asse X che dell'asse Y, aumentando la complessiva durevolezza e precisione del touch-screen.

SENSORI TATTILI CAPACITIVI SUPERFICIALI

In confronto, la tecnologia tattile capacitiva è in grado di offrire qualità ottiche superiori. Inoltre, non è necessario uno strato flessibile, per cui gli schermi capacitivi possono essere resi altamente duraturi e resistenti alle condizioni ambientali. La tecnologia tattile capacitiva superficiale si distingue dalla tecnologia capacitiva proiettata in quanto il substrato è rivestito solo da un unico strato conduttivo uniforme. Il conduttore è protetto da strati supplementari di materiale isolante, ma lo spessore è limitato a causa del compromesso con la sensibilità tattile.
Esistono numerosi e diversi metodi di rilevamento, utilizzati nell'industria, ma tutti sono basati sul principio di misurazione dell'aumento di capacitanza che si registra quando un dito o uno stilo conduttivo viene avvicinato allo strato conduttivo.
I pulsanti e i dispositivi di scorrimento sensibili capacitivi non necessariamente devono essere trasparenti, per cui è potenzialmente possibile utilizzare una vasta gamma di materiali per lo strato conduttivo (ad esempio il tipico rame PCB). Inoltre, è disponibile un'ampia gamma di tecniche di rilevamento. Ad esempio, le modifiche della capacitanza possono essere rilevate applicando allo strato conduttivo una tensione variante nel tempo e misurando la variazione del comportamento RC del circuito. I metodi di trasferimento del carico costituiscono un'altra opzione; l'aumento della capacitanza del sensore a seguito del tocco con un dito determina un maggiore trasferimento del carico elettrico a un condensatore di riferimento, che viene misurato come il potenziale che lo attraversa. Mentre i pulsanti tattili sono realizzati a partire da un singolo pad conduttivo, i dispositivi di scorrimento sono realizzati come una disposizione in serie di più pad. I design capacitivi superficiali di un touch-screen sono leggermente più limitati. Per lo strato conduttivo vengono scelti ossido di indio-stagno (ITO) o analoghi ossidi conduttivi trasparenti, dato che la trasparenza del sensore è un requisito comune. Gli elettrodi sono disposti con precisione attorno allo strato conduttivo per generare una tensione in corrente alternata uniforme attraverso l'intero foglio. Quando un dito, o uno stilo conduttivo, viene posizionato in un punto dello schermo, l'utente diviene un percorso ad accoppiamento capacitivo verso massa. Il risultante prelievo di corrente viene quindi misurato, in genere, da ogni angolo del foglio capacitivo e i valori vengono utilizzati per calcolare le coordinate X e Y della posizione tattile.
Sebbene i design capacitivi superficiali offrano una migliore trasmissione ottica rispetto agli schermi resistivi e non siano caratterizzati dalla maggiore complessità degli schermi capacitivi proiettati, possono essere soggetti ad accoppiamento capacitivo parassitico e presentano, in generale, la risoluzione più scarsa fra le tre tipologie. I touch-screen capacitivi superficiali sono idonei per applicazioni quali chioschi multimediali e comandi industriali di base.

Sensori tattili capacitivi proiettati

La maggior parte dei sensori touch-screen a capacitanza proiettata consistono in due strati conduttivi trasparenti, in genere di ossido di indio-stagno (ITO), separati da un sottile isolante per formare uno schema a griglia. Alcuni dei più recenti design sottili lo realizzano in un unico strato, polverizzando la griglia in ITO su un unico substrato.
Nell'industria degli smartphone, i touch-screen capacitivi proiettati stanno diventando sempre più diffusi grazie a una gamma di vantaggi rispetto alle altre tecnologie tattili, tra cui lunga durata di vita, elevata durevolezza, qualità ottiche superiori e sensibilità multi-touch reale. Lo schermo capacitivo proiettato di un tipico smartphone può possedere qualche centinaia di intersezioni di elettrodi; tuttavia è possibile ottenere una risoluzione molto più elevata tramite speciali algoritmi

che interpolano i valori degli elettrodi adiacenti.
La tecnologia tattile proiettata può essere ulteriormente classificata in schermi auto-capacitivi o capacitivi reciproci.

Schermi auto-capacitivi

Gli schermi auto-capacitivi rilevano il tocco misurando la somma della (auto-)capacitanza parassitica e della capacitanza aggiunta da un dito o da uno stilo conduttivo. In termini di sofisticazione, questo metodo si colloca all'incirca a metà strada tra i touch-screen resistivi e gli schermi di tipo capacitivo reciproco più complessi.
Gli elettrodi sono distribuiti in una griglia a due strati e le misurazioni sono effettuate non a partire da una intersezione di elettrodi, ma da una coppia di elettrodi a riga e colonna. Ciò rappresenta un problema per la funzionalità multi-touch. Quando lo schermo viene toccato in due diverse posizioni, esistono due serie di possibili punti che possono essere interpretati dal dispositivo tattile (ciò forma un rettangolo con ciascuna coppia su una diagonale), senza alcuna possibilità per il software di stabilire quale coppia corrisponde alle attuali posizioni toccate. Le due posizioni false sono chiamate "punti fantasma". Fortunatamente molti gesti multi-touch (quali ad esempio il movimento di avvicinamento delle dita) non sono sensibili a questo effetto **(figura 3)**.

Sebbene molti design più recenti, in particolare per gli smartphone, si siano indirizzati verso i touch-screen capacitivi reciproci, la tecnologia auto-capacitiva è tuttora in uso e continua a migliorare.

Schermi capacitivi reciproci

I sensori capacitivi proiettati reciproci rappresentano una tecnologia più recente e misurano la capacitanza tra elettrodi (anziché da elettrodo a massa). Pertanto, a differenza dei touch-screen auto-capacitivi, le misurazioni possono essere effettuate in corrispondenza di ogni intersezione di elettrodi. Le intersezioni vengono "scansionate" singolarmente, in genere movimentando un elettrodo a colonna, quindi scansionando rapidamente ogni elettrodo a riga che lo interseca. Ciò risulta vantaggioso poiché significa che la struttura hardware supporta un rilevamento non ambiguo di tocchi simultanei pressoché illimitati.
La capacitanza reciproca consente inoltre una gamma più vasta di design a schema di sensori, che contribuisce a massimizzare il rapporto segnale/rumore (SNR). Ciò risulta vantaggioso poiché le capacitanze misurate possono trovarsi nella gamma femtofarad, rendendo il rumore dell'LCD particolarmente problematico.
La tecnologia capacitiva reciproca sta comunque gradualmente sostituendo la sua controparte ad auto-capacitanza in numerose applicazioni, essendo in generale più robusta e otticamente meno ostruttiva.

UNO SGUARDO AL FUTURO

Quelle appena descritte non sono le uniche tecnologie tattili attualmente in uso. Ne esistono molte altre, ciascuna con specifici vantaggi. Tuttavia, la tecnologia e l'innovazione continuano a compiere grandi passi avanti: sono attualmente in progettazione touch-screen capacitivi in grado di funzionare anche se si indossano guanti e presto assisteremo all'introduzione di display capacitivi proiettati con funzionalità di rilevamento sensibile al passaggio del puntatore. Nel frattempo, Mouser Electronics continua a mantenere la propria leadership, offrendo prodotti e strumenti di ultima generazione nell'intento di accelerare la crescita di questo fiorente settore industriale. ❑

CODICE MIP 2828832

Corso mikroBASIC

Utilizzare un RICEVITORE GPS CON IL PIC

In questa puntata del corso sui microcontrollori faremo finalmente chiarezza sull'utilizzo dei ricevitori GPS e sulla programmazione dei PIC per determinare al meglio la nostra posizione sul globo terrestre.

IL GPS

GPS è l'acronimo delle parole inglesi "Global Positioning System". Questo sistema per la ricezione dei dati di posizionamento è estremamente affascinante. Il pensiero che alcuni satelliti, posizionati a migliaia di chilometri dal suolo terrestre, possano determinare la posizione e l'altitudine con una precisione straordinaria, a volte anche al decimetro, rende stuzzicante qualsiasi tipologia di progetto. Ma con il GPS non si rileva solo la propria posizione e la propria altitudine. I dati ricevibili sono infatti molto numerosi e di estrema importanza. È possibile infatti rilevare l'ora corrente (con la precisione degli orologi atomici), il numero di satelliti utilizzati, le informazioni sulla trasmissione, la nostra velocità di crociera con la re-

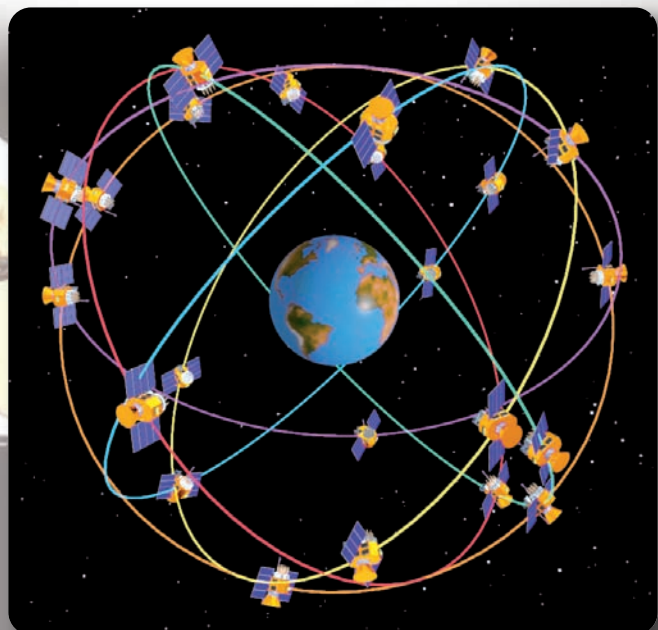



Figura 1: la costellazione dei satelliti GPS.

Table 1-1 NMEA Output Messages

Message	Description
GGA	Time, position and fix type data
GLL	Latitude, longitude, UTC time of position fix and status
GSA	GPS receiver operating mode, satellites used in the position solution, and DOP values
GSV	Number of GPS satellites in view satellite ID numbers, elevation, azimuth, & SNR values
MSS	Signal-to-noise ratio, signal strength, frequency, and bit rate from a radio-beacon receiver
RMC	Time, date, position, course and speed data
VTG	Course and speed information relative to the ground
ZDA	PPS timing message (synchronized to PPS)
150	OK to send message
151	GPS Data and Extended Ephemeris Mask
152	Extended Ephemeris Integrity
154	Extended Ephemeris ACK

Figura 2: i messaggi NMEA.

lativa direzione, e altri dati molto interessanti. Il metodo seguito per raggiungere tanta precisione è quello della triangolazione. Più satelliti sono in uso, maggiore sarà la precisione di misura ottenuta.
Ovviamente questa non è la sede adatta per descrivere in dettaglio le modalità di trasmissione, le sincronizzazioni e ogni altra informazione tecnica che sta alla base della materia. L'articolo ha invece lo scopo di far comprendere come programmare il microcontrollore per la ricezione dei dati, come decodificarli e infine come visualizzarli. Esso inoltre ha l'obiettivo di fornire un orientamento base ai principianti, affinché essi possano approcciare la ricezione GPS in maniera semplice e indolore, per poi affinare le tecniche di programmazione.

NMEA

La raccolta, l'analisi e la programmazione dei dati raccolti tramite un ricevitore GPS è piuttosto semplice. Un microcontrollore pone qualche difficoltà maggiore, rispetto a un PC, per via dei suoi limiti strutturali. Questo implica una maggiore applicazione e una codifica superiore, in dipendenza anche dal linguaggio di programmazione utilizzato e dal tipo di compilatore adottato.
Le informazioni ricevute da un ricevitore GPS possono a volte essere restituite all'utente attraverso una sequenza di caratteri. Tale insieme di dati è detta stringa NMEA (National Marine Electronics Association). Il programmatore può estrarre questa successione di caratteri, per poi esaminare le varie parti che la compongono.
Tale standard è stato introdotto dalla Marina americana.
Un esempio di ricezione NMEA (log NMEA) è visibile nel **riquadro 1**:
Come si nota, ogni stringa, se correttamente ricevuta e processata, inizia sempre con il carattere di dollaro ("$"). Solitamente il processo di analisi si completa sino al raggiungimento del carattere asterisco ("*"). I vari parametri ricevuti sono separati tra loro da una virgola (","). Per la comprensione del formato di trasmissione e del significato di ogni parametro, si rimanda il lettore alla letteratura ufficiale (NMEA Reference Manual), facilmente reperibile su Internet.

Messaggi e sentenze

Ogni riga ricevuta inizia sempre, come visto, dalla sequenza "$GP". Si tratta di tre byte che attestano l'inizio di un messaggio, da parte del ricevitore GPS. A questi tre byte seguono sempre altri tre byte che specificano la tipologia di informazione spedita. L'utente può quindi "filtrare" il messaggio, secondo le proprie necessità. Non è necessario infatti processare tutte le sentenze ma basta solo elaborare quella che più interessa.
Nell'esempio visto sopra, possiamo notare, in corrispondenza del quarto, quinto e sesto carattere, le sentenze "RMC", "VTG", "GSA", "GSV" e "GLL".
Una volta scelta e isolata la stringa con la funzionalità desiderata, è possibile separare i vari parametri che sono, ricordiamolo, separati da virgole.

IL RICEVITORE SMARTGPS

Per i nostri esperimenti abbiamo adottato la scheda SmartGPS, prodotta da MikroElektronika. Ovviamente può essere utilizzato qualsiasi altro ricevitore con standard NMEA, sia della stessa azienda di altri produttori. Logicamente, in questi casi, occorre apportare le dovute modifi-

Riquadro 1

```
$GPRMC,165030.00,A,3729.87492,N,01403.20141,E,0.532,,040912,,,A*74
$GPVTG,,T,,M,0.532,N,0.984,K,A*22
$GPGGA,165030.00,3729.87492,N,01403.20141,E,1,10,0.91,680.7,M,37.6,M,,*50
$GPGSA,A,3,06,22,19,16,14,11,01,30,32,21,,,1.45,0.91,1.12*04
$GPGSV,3,1,12,01,10,269,27,03,87,262,08,06,76,124,32,11,21,284,14*7B
$GPGSV,3,2,12,14,22,124,20,16,35,202,31,18,28,047,17,19,55,318,13*7E
$GPGSV,3,3,12,21,07,075,23,22,68,070,19,30,10,183,35,32,09,212,25*7C
$GPGLL,3729.87492,N,01403.20141,E,165030.00,A,A*67
```

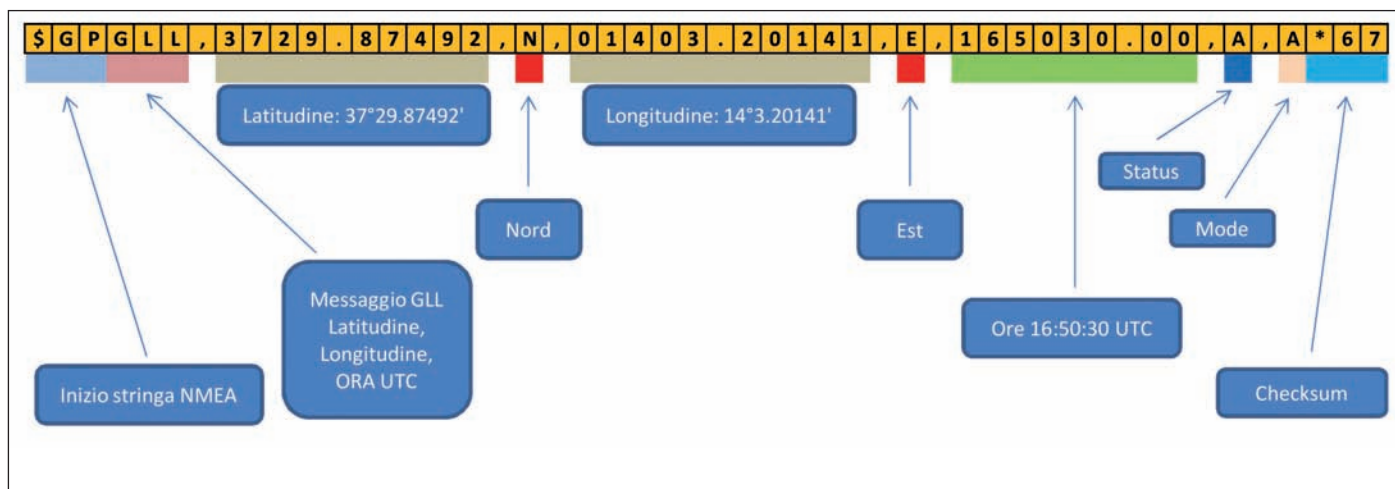



Figura 3: una stringa NMEA ricevuta di tipo GLL.

che allo schema elettrico e, soprattutto, alle connessioni tra PIC e RX.
Il ricevitore SmartGPS è ottimo per sperimentare la ricezione GPS, poiché può essere utilizzato sia in abbinamento a un microcontrollore (PIC, dsPIC, AVR e ARM), sia indipendentemente, collegandolo direttamente alla porta di un PC. È infatti possibile, installando il suo driver Blox, identificare la scheda come una porta seriale virtuale (COM) e aprirla all'occorrenza con il linguaggio di programmazione preferito.
Successivamente pubblicheremo un articolo proprio su questo argomento.
La scheda SmartGPS è in grado di acquisire facilmente dati dai satelliti con pochi settaggi. Il funzionamento è basato sul modulo LEA-6S. Esso dispone di alcuni jumper e switch con i quali, facilmente, si possono configurare diverse operazioni. La comunicazione avviene tramite UART oppure USB. Si può decidere se la comunicazione debba avvenire con logica a 3,3 V oppure a 5 V. Dispone inoltre di una batteria on board e si la può collegare a un'antenna passiva o attiva.

Posizione degli switch

Per utilizzare al meglio lo SmartGPS è opportuno configurare correttamente gli switch e i jumper presenti sulla scheda. Ecco, in dettaglio, la loro funzione:

- SW1: è un DIP formato da 8 microinterruttori. Si usa per selezionare il tipo di microcontrollore utilizzato. Per il PIC occorre spostare gli interruttori 1 e 5 in posizione ON e tutti gli altri in posizione OFF;
- J2: specifica il tipo di antenna utilizzata. È possibile scegliere tra passiva e attiva;
- J1: seleziona la modalità di alimentazione elettrica. Le scelte possono essere USB oppure esterna;
- J3: permette di scegliere la tensione di alimentazione, tra i valori possibili di 3,3 V e 5 V;
- J9: abilita il LED lampeggiante che indica la ricezione satellitare;
- J8: configurazione.

SCHEMA ELETTRICO

Lo schema elettrico del collegamento tra PIC e SmartGPS è visibile in figura. Si notano tre unità distinte:

- il microcontrollore PIC 18F452;
- il display grafico GLCD con risoluzione 128x64 pixel;
- il ricevitore SmartGPS.

Il PIC potrebbe essere sostituito con un altro modello, purché quello abbia abbastanza memoria da contenere il firmware e, soprattutto, che sia dotato di supporto UART. Esso è fatto oscillare grazie alla presenza del quarzo (anche da 4 Mhz) e dai due condensatori ceramici da 22 pF.
La PORTD è interamente dedicata al trasporto dati verso il display grafico, mentre ben sei porte della PORTB trasportano i segnali di controllo del GLCD.
La comunicazione con lo SmartGPS avviene attraverso il canale UART del PIC18F4520. La porta RC6 invia dati verso il ricevitore GPS mentre la porta RC7 è posta in ricezione.
Per i nostri scopi pratici è sufficiente solamente ricevere dati, per cui è esclusivamente utilizzata la porta RC7.
Per un aspetto estetico migliore è possibile usare il connettore piatto che collega la EasyPIC allo SmartGPS ma, come abbiamo detto, i collegamenti possono fare capo solo ai piedini VCC, GND e P7 dello SmartGPS.

CONNESSIONE EASYPIC E SMARTGPS

La connessione avviene attraverso la PORTC. Allo scopo, è sufficiente collegare un cavo piatto a pettine formato da 10 pin

Figura 4: lo SmartGPS.

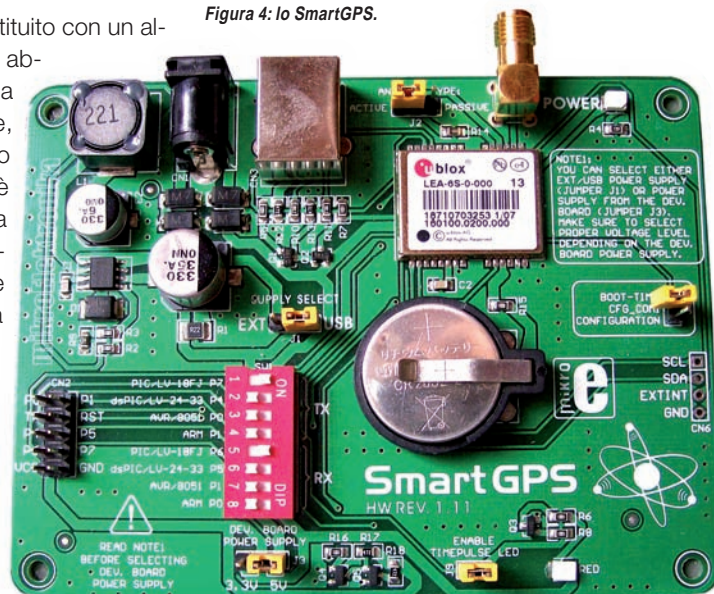
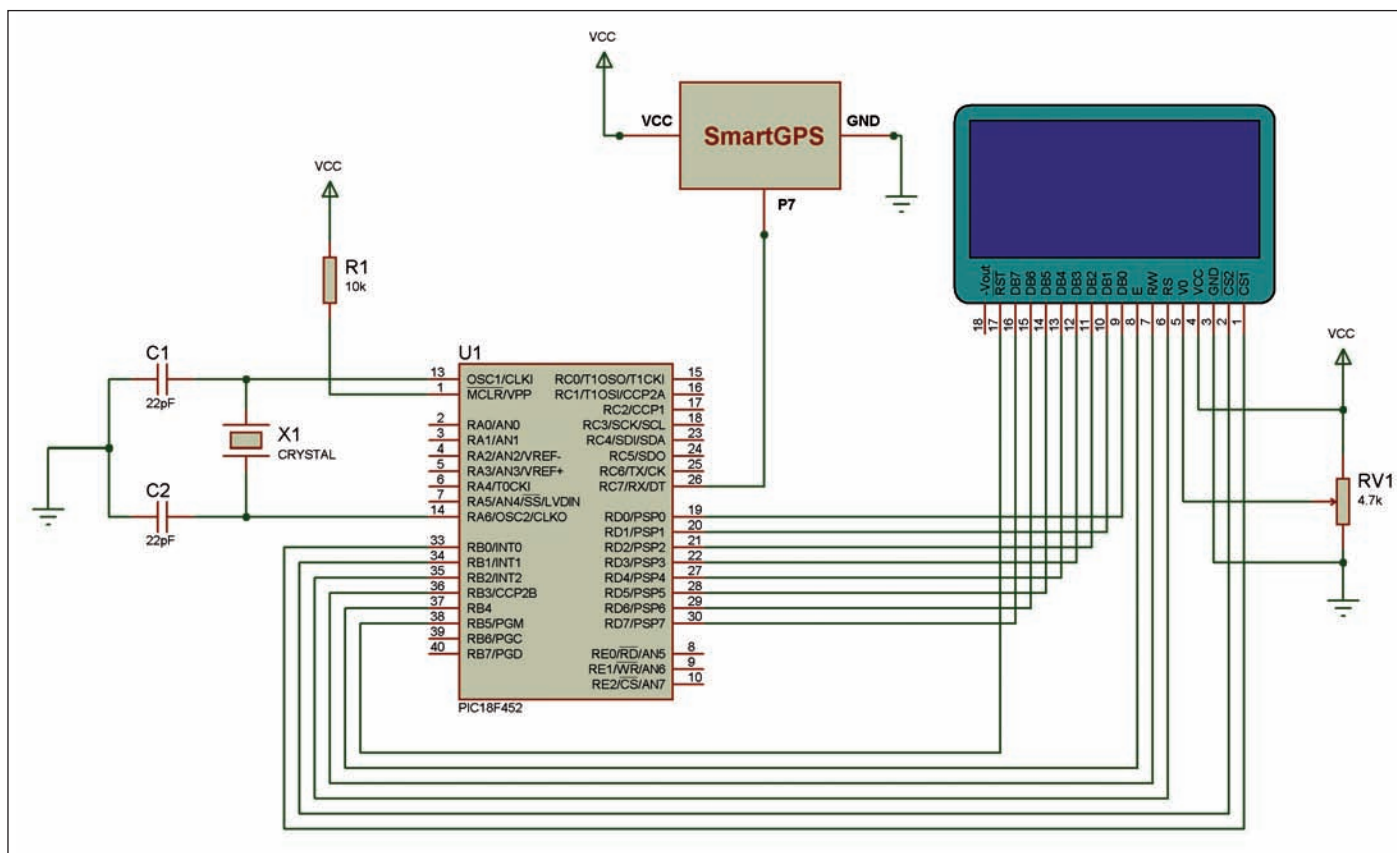



Figura 5: schema elettrico.

DIL (5x2). In alternativa, tramite tre fili con connettori singoli, è possibile unire i seguenti piedini:

- VCC della EasyPIC con VCC dello SmartGPS;
- GND della EasyPIC con GND dello SmartGPS;
- RC7 della EasyPIC con P7 dello SmartGPS.

È importante che tra i due moduli vi sia sempre la massa in comune. Solo in questo modo è infatti possibile la trasmissione corretta dei dati. Per il corretto funzionamento con la EasyPIC, è necessario inoltre:

1. Disattivare le resistenze di Pull-Up e Pull-Down della PORTC, sulla EasyPIC. Esse infatti farebbero sorgere problemi alla linea UART;
2. Disattivare anche i diodi LED collegati alla PORTC. Anch'essi renderebbero impossibile una fluida comunicazione UART;
3. Attivare il display GLCD.

È ovvio che non è necessario utilizzare il display grafico. Al suo posto può essere infatti adoperato il display testuale LCD, molto meno performante, in quanto possiede meno spazio logico per i messaggi, ma altrettanto efficace. In questo caso occorre modificare il firmware affinché esso gestisca tale tipologia di visualizzazione.

IL PROGRAMMA

Dedichiamoci adesso al programma. Non è estremamente complicato ma nemmeno troppo semplice. Trattando le stringhe NMEA occorre seguire un procedimento idoneo ed efficace. L'idea algoritmica di base è la seguente:

- si legge un carattere (se disponibile) dalla porta UART sino alla ricezione di un carattere '$' (dollaro, codice ASCII 36). Questo simbolo testimonia l'inizio di una "sentence" e, in generale, di una linea NMEA;
- si inizia a memorizzare, sequenzialmente e in un vettore, tutti i caratteri successivi;
- si continua ad acquisire i caratteri e ad accodarli al vettore, fino a quando si incontra il carattere '*' (asterisco, codice ASCII 42). Questo carattere segna dunque la fine logica della stringa, dal momento che dopo di esso c'è solo il checksum.

L'esecuzione di questo algoritmo porta alla generazione di un array a indice numerico (che inizia da 0) contenente la stringa NMEA. Un esempio di array di questo tipo si può vedere nel **riquadro 2**: Dall'esempio possiamo evincere che la sentenza GLL si trova in posizione 2, 3 e 4, il punto cardinale 'N' (nord) si trova alla posizione 17 e la longitudine si trova a partire dalla posizione 19 fino alla virgola successiva.

Quindi la difficoltà nella programmazione non sta tanto nella ricezione e nella costruzione della stringa NMEA, quanto nel separare i vari elementi che la compongono.

Ricordiamo inoltre che le "sentence" delle stringhe NMEA sono molteplici, e ogni caso particolare va approfondito e codificato singolarmente. Inoltre il mikroBasic non dispone di semplici funzioni ad alto livello che gestiscano in autonomia le strin-

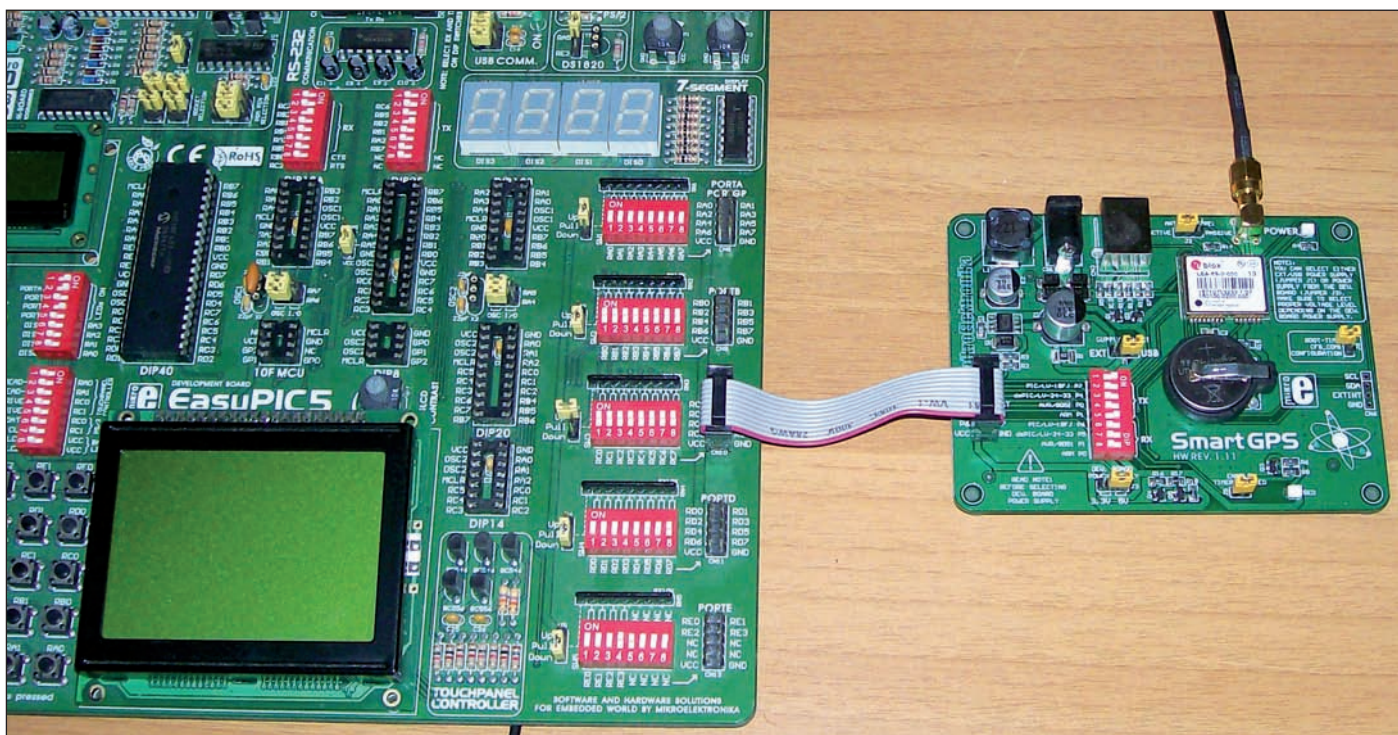



Figura 6: connessione tra EasyPIC e SmartGPS con cavo piatto.

ghe. Il programmatore deve dunque scrivere da sé tali funzioni, al fine di determinare la posizione di sotto-stringhe e di separare i vari elementi.

Il listato

Il sorgente del programma (scaricabile dal sito di *Fare Elettronica*) merita di essere studiato e commentato in ogni sua parte. Il listato contiene numeri sequenziali a ogni riga. Essi non devono essere digitati nell'editor del mikroBasic ma servono unicamente quale riferimento descrittivo. Ecco, in dettaglio, i commenti relativi a ogni blocco logico di istruzioni. Consigliamo di capire a fondo il flusso e il significato delle varie parti che lo compongono. Il vero programmatore, infatti, non si limita a effettuare un semplice copia e incolla dei listati trovati in Rete, ma dovrebbe completarli con proprie integrazioni o, meglio ancora, scrivere il firmware integralmente.

- le linee da 9 a 22 configurano le porte del display GLCD e, precisamente, definiscono la linea dati e la linea comandi;
- le linee da 24 a 29 servono per la dichiarazione delle variabili utilizzate;
- le linee da 32 a 36 definiscono la funzione "ReadChr()" che ha lo scopo, quando invocata, di restituire il valore di un carattere letto dalla UART, quando esso è disponibile sul buffer di linea;
- la linea 38 definisce l'inizio del programma principale;
- le linee da 40 a 43 settano il display grafico;
- la linea 45 regola la velocità della linea UART1 a 9600 baud. Essa fa capo alle porte RC6 e RC7 del PIC;
- le linee comprese tra la 47 e la 127 racchiudono un loop infinito, nel quale il programma è eseguito in modo continuo;
- le linee da 48 a 51 attendono l'arrivo di un carattere dollaro ('$', Ascii 36) che testimonia l'inizio di una linea NMEA. Appena esso è presente sulla linea, il loop ha termine e il programma continua con le successive istruzioni;
- le linee tra 53 a 59 inseriscono e accodano in un array il valore dei caratteri acquisiti dal ricevitore, costruendo una stringa NMEA, gestibile dal programmatore;
- le linee comprese tra 61 e 126 sono eseguite se la sentenza ricevuta corrisponde a "GLL". Per verificare ciò basta controllare il terzo, quarto e quinto carattere (ri-

Riquadro 2

```
-------------------------------------------------------------
                    1         2         3         4
posizione: 0123456789012345678901234567890123456789012345678
-------------------------------------------------------------
NMEA:      GPGLL,3729.87213,N,01403.20040,E,165028.00,A,A*61
-------------------------------------------------------------
```

cordiamo che il carattere '$' non viene memorizzato nel nostro vettore e che l'indice del primo carattere è 0);

- le linee comprese tra 64 a 72 hanno il compito di trovare il punto di inizio della latitudine, che si trova un carattere dopo la prima virgola incontrata durante la scansione del vettore;
- le linee comprese tra 73 a 79 hanno il compito di trovare il punto finale della latitudine, che si trova un carattere prima della successiva virgola;
- le linee comprese tra 80 a 90 costruiscono, carattere dopo carattere, una nuova stringa (cLatitudine), contenente le informazioni sulla latitudine. La stringa è terminata con un carattere ASCII(0), per definirne la terminazione;
- la riga 92 estrae il punto cardinale della latitudine (nord o sud);
- le linee comprese tra 94 a 102 hanno il compito di trovare il punto finale della longitudine, dal momento che il punto iniziale si trova indirettamente con la precedente operazione;
- le linee comprese tra 103 a 113 costruiscono, carattere dopo carattere, una nuova stringa (cLongitudine), contenente le informazioni sulla longitudine. La stringa è terminata con un carattere ASCII(0), per definirne la terminazione;
- la riga 115 estrae il punto cardinale della longitudine (est o ovest);
- le linee comprese tra 117 a 125 hanno infine lo scopo di visualizzare le informazioni sul display grafico.

Figura 7: connessione manuale tra EasyPIC e SmartGPS con tre fili.

Metodo alternativo di analisi stringa

Un metodo ancora migliore per analizzare una stringa NMEA ,e costruire tante sotto-stringhe utilizzabili, è il seguente: si legge un carattere dalla UART e si accoda in una stringa (ad esempio cLatitudine). Appena si trova una virgola, si esclude quest'ultima e si accodano i successivi caratteri acquisiti in una nuova stringa, e cosi via, fino alla fine della stringa principale. In tal modo saranno costruite tante stringhe di caratteri, ognuna delle quali conterrà le informazioni desiderate. Se devono essere visualizzate, occorre terminare le stringhe con il carattere chr(0).

Figura 8: il firmware in azione.

CONCLUSIONI

Riuscire a gestire in proprio le stringhe NMEA è un'operazione che regala davvero tante soddisfazioni. Collaudare il prototipo con la propria autovettura in movimento e accorgersi che il display fornisce coordinate sempre diverse è un'emozione molto appagante.

Naturalmente le informazioni che abbiamo fornito nel presente articolo sono solo il minimo indispensabile per creare un'applicazione pratica più complessa. Soffermiamoci su alcune considerazioni. Le coordinate ricevute con gli esempi studiati sono semplici stringhe. Per poterle utilizzare praticamente, in calcoli o misurazioni reali, occorre trasformarle in valori numerici a precisione decimale. Inoltre, come si evince dal riferimento ufficiale NMEA, le posizioni sul globo terrestre sono fornite nel formato ddmm.mmmm (latitudine) oppure dddmm.mmmm (longitudine).

Questo formato restituisce un valore in gradi e minuti (con frazione di minuti). Se occorre un riferimento espresso com-

Figura 9: algoritmo generico per l'acquisizione di una stringa NMEA.

GGA —Global Positioning System Fixed Data

Table 1-3 contains the values for the following example:

$GPGGA,002153.000,3342.6618,N,11751.3858,W,1,10,1.2,27.0,M,-34.2,M,,0000*5E

Table 1-3 GGA Data Format

Name	Example	Unit	Description
Message ID	$GPGGA		GGA protocol header
UTC Time	002153.000		hhmmss.sss
Latitude	3342.6618		ddmm.mmmm
N/S Indicator	N		N=north or S=south
Longitude	11751.3858		dddmm.mmmm
E/W Indicator	W		E=east or W=west
Position Fix Indicator	1		See Table 1-4
Satellites Used	10		Range 0 to 12
HDOP	1.2		Horizontal Dilution of Precision
MSL Altitude	27.0	meters	
Units	M	meters	
Geoid Separation	-34.2	meters	Geoid-to-ellipsoid separation. Ellipsoid altitude = MSL Altitude + Geoid Separation.
Units	M	meters	
Age of Diff. Corr.		sec	Null fields when DGPS is not used
Diff. Ref. Station ID	0000		
Checksum	*5E		
<CR> <LF>			End of message termination

Table 1-4 Position Fix Indicator

Value	Description
0	Fix not available or invalid
1	GPS SPS Mode, fix valid
2	Differential GPS, SPS Mode, fix valid
3-5	Not supported
6	*Dead Reckoning Mode, fix valid*

Figura 10: formato stringa di tipo GGA.

GLL—Geographic Position - Latitude/Longitude

Table 1-5 contains the values for the following example:

$GPGLL, 3723.2475,N,12158.3416,W,161229.487,A,A*41

Table 1-5 GLL Data Format

Name	Example	Unit	Description
Message ID	$GPGLL		GLL protocol header
Latitude	3723.2475		ddmm.mmmm
N/S Indicator	N		N=north or S=south
Longitude	12158.3416		dddmm.mmmm
E/W Indicator	W		E=east or W=west
UTC Time	161229.487		hhmmss.sss
Status	A		A=data valid or V=data not valid
Mode	*A*		*A=Autonomous, D=DGPS, E=DR (Only present in NMEA v3.00)*
Checksum	*41		
<CR> <LF>			End of message termination

Figura 11: formato stringa di tipo GLL.

pletamente in gradi (decimali), occorre eseguire le opportune conversioni matematiche.
Quando si utilizza lo SmartGPS, come per qualsiasi altro tipo di ricevitore GPS, occorre attendere qualche minuto, prima che esso riceva i segnali. Lo stato di funzionamento è testimoniato dal lampeggiare del diodo LED presente su di esso, se attivato (Timepulse Led).
Ricordarsi, infine, di configurare correttamente i flag di programmazione del PIC, pena il mancato funzionamento del firmware. Di solito è meglio disabilitare tutte le opzioni.
Con le specifiche viste nell'articolo, e tenendo sott'occhio la documentazione ufficiale NMEA, si invita il lettore a realizzare un semplice altimetro, per misurare l'altitudine del proprio sito rispetto al livello del mare. ❏

CODICE MIP 2828840

CAMPAGNA ABBONAMENTI

PROMO tutte 3

RISPARMIO ASSICURATO!
Risparmierai fino a 73,00 EUR sul prezzo di copertina. Garanzia del prezzo bloccato per un anno. Le riviste ti saranno recapitate comodamente a casa. Riceverai un buono sconto del 20% per un tuo prossimo acquisto su **www.ieshop.it**

UNA PROMOZIONE ESCLUSIVA:

- 11 numeri di FARE ELETTRONICA
- 11 numeri di ELEKTOR
- 11 numeri di FIRMWARE (edizione digitale) compreso l'accesso all'archivio a partire dal numero 49 di febbraio 2010
- 1 anno di accesso al CLUB di FARE ELETTRONICA
- 1 anno di accesso al CLUB di FIRMWARE,
- 20% di sconto per un acquisto su www.ieshop.it

TUTTO A SOLI € 119,50 ANZICHE' € 192,50

Sì, mi abbono alla PROMOTUTTE3 (è possibile abbonarsi online www.ieshop.it/promotutte3)

Riceverò 11 numeri di Fare Elettronica e 11 di Elektor in edizione cartacea, potrò accedere all'archivio di Firmware in edizione digitale a partire dal numero 49 di febbraio 2010 e ai prossimi 11 numeri, compreso un anno di accesso al CLUB di Fare elettronica e Firmware, tutto a **soli € 119,50** anziché €192,50

METODI DI PAGAMENTO

☐ **BONIFICO BANCARIO** ☐ **BOLLETTINO POSTALE** ☐ **ALLEGO ASSEGNO** (Intestato a Inware Edizioni)

☐ MasterCard ☐ VISA ☐ postepay

PayPal fare il pagamento su paypal@inwaredizioni.it

TITOLARE. . . . .
NUMERO . . . .
DATA DI SCADENZA. . . . . CODICE DI SICUREZZA . . . .

INVIARE IL MODULO D'ORDINE A: Inware Edizioni srl, via Giotto 7 - 20032 Cormano (MI) - oppure via FAX allo 02 66508225
(Inviando questo modulo si accettano i termini e le condizioni riportate a tergo)

Nome . . . . Cognome . . . .
Indirizzo . . . . n° . . . . Cap . . . .
. . . . . Città . . . . Prov . . . .
Tel . . . . Fax . . . . Email . . . .
Ragione Sociale . . . . Codice fiscale . . . . Partita IVA . . . .
DATA . . . . FIRMA . . . . ☐ Fattura

1 PERCHÉ SCEGLIERE FARE ELETTRONICA:

LAVORO: Se lavori come progettista elettronico, Fare Elettronica saprà esserti utilissima con numerose idee e spunti sia dal punto di vista pratico che teorico. DIDATTICA: se lavori come insegnante in una scuola ad indirizzo tecnico, Fare Elettronica risulterà tra le migliori fonti di ispirazione per te e per i tuoi allievi. AGGIORNAMENTO TECNOLOGICO: se sei alla ricerca di una fonte d'ispirazione, Fare Elettronica non ti deluderà, fornendoti l'aggiornamento sulle più recenti tecnologie, trattate in modo esauriente e competente. HOBBY: se invece l'elettronica è la tua passione, Fare Elettronica ti permetterà di coltivarla giorno per giorno con progetti sempre appassionanti ed alla tua portata.

2 PERCHÉ SCEGLIERE FIRMWARE:

Firmware è una fantastica rivista digitale, tra le prime uscite nel settore dell'elettronica e la prima in assoluto in lingua italiana. Nasce come naturale evoluzione della rivista Firmware in edizione cartacea, rivista dedicata ai professionisti dell'elettronica, esigenti e competenti bisognosi di uno strumento di aggiornamento e di know how per l'elettronica professionale. Firmware è basata su una piattaforma che saprà regalarti una nuova esperienza di accesso a incredibili contenuti. Nessuna rivista ti sembrerà più all'altezza dopo aver visto Firmware!

3 PERCHÉ SCEGLIERE ELEKTOR:

Elektor è in circolazione in oltre 50 Paesi con un totale di 300,000 copie mensili. Pubblicato in Inghilterra, Francia, Paesi Bassi, Spagna, Grecia, Portogallo, Brasile, Svezia, Finlandia, Germania, Italia, Stati Uniti. La rivista Elektor è distribuita a livello mondiale tramite abbonamento, edicole e librerie. I siti internet sono tradotti in Inglese, Tedesco, Francese, Italiano, Cinese, Portoghese, Brasiliano, Svedese e Greco. 2 milioni di lettori mensili. 140.000 visitatori web mensili. 43.000 abbonati a e-weekly, la newsletter digitale. Elektor propone progetti di elevata qualità ideali per l'autocostruzione nei campi dei microcontrollori, audio high-end, RC modelling, RF e test & measurement. Elektor è creata da veri appassionati di elettronica. Elektor è l'unica rivista internazionale indipendente a vantare un laboratorio di progettazione e PCB design interno.

PERCHÉ SCEGLIERE TUTTE E TRE LE RIVISTE

Perché solo Inware Edizioni può darti una offerta così completa a condizioni così convenienti con un mix di aggiornamento, conoscenza, stimolo, divertimento e professionalità! Approfitta della fantastica PROMOTUTTE3 non te ne pentirai

CODICE MIP 2828943

di GIUSEPPE LA ROSA

Temporizzatore per BROMOGRAFO

Realizziamo un bromografo con temporizzatore digitale gestito da microcontrollore e accensione elettronica delle lampade.

In questo articolo ci occuperemo di un semplice temporizzatore per bromografo, programmabile da 1 a 3600 secondi, con la visualizzazione del tempo impostato e del tempo trascorso su display LCD 16x2 con retroilluminazione a spegnimento automatico soft.
Inoltre il temporizzatore dispone di un buzzer che segnala con un suono la fine del tempo di esposizione delle piastre. Il buzzer viene attivato anche come tono tasti e al superamento delle soglie 1 e 3600 secondi, eseguendo l'Error beep il quale, per mezzo di un suono lungo, indica il limite raggiunto. L'apparechio è utile in laboratorio per la realizzazione di circuiti stampati e la cancellazione di memorie UVA.

SCHEMA ELETTRICO

Lo schema elettrico visibile in **figura 2** è piuttosto semplice. Il circuito è basato sul microcontrollore IC1 (PIC16F628) che

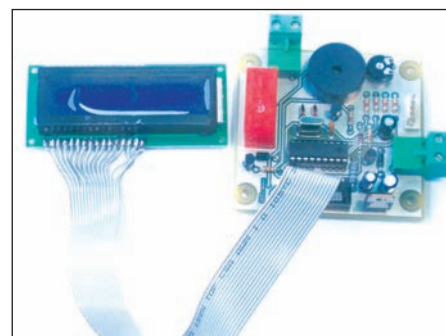



Figura 1: fotografia del progetto finito.

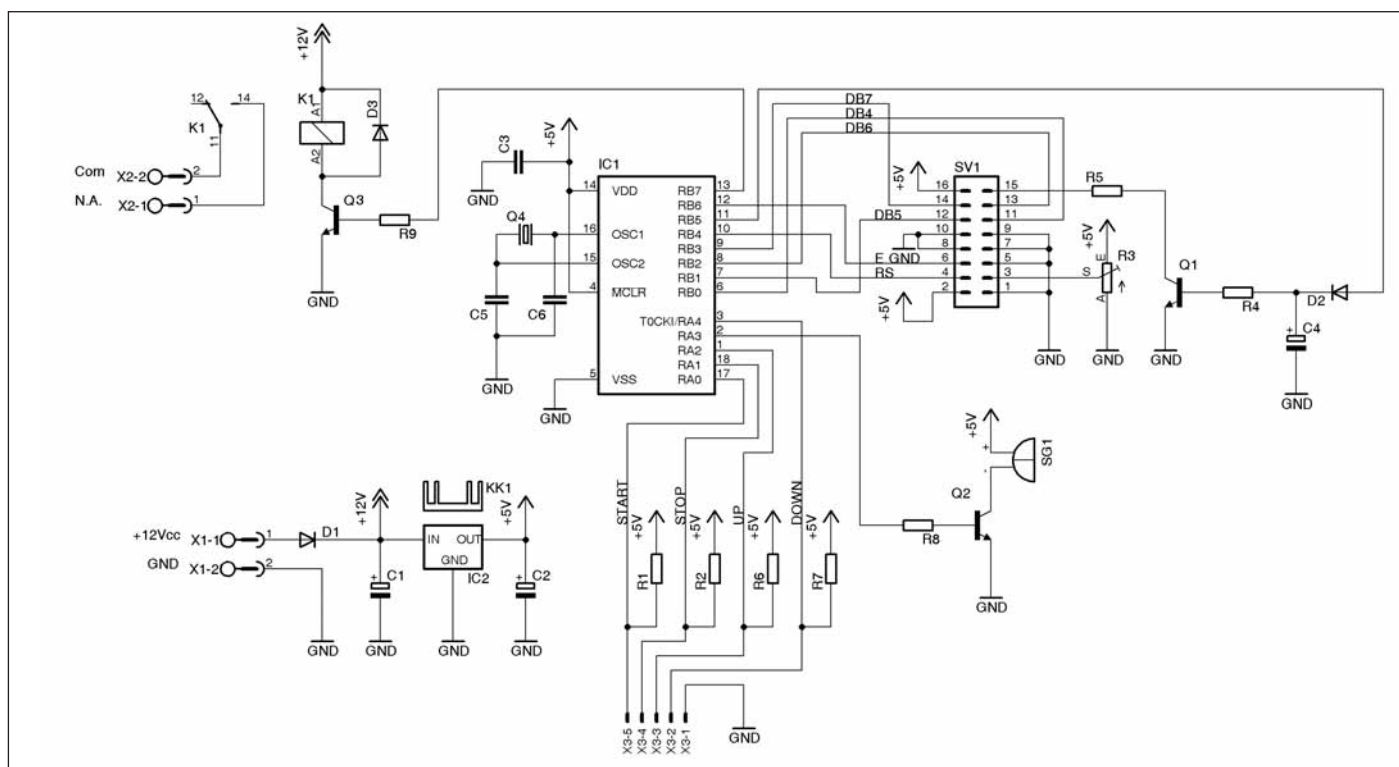



Figura 2: schema elettrico del temporizzatore.

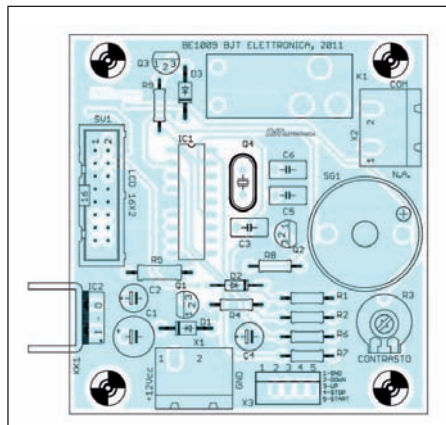



Figura 3: piano di montaggio.

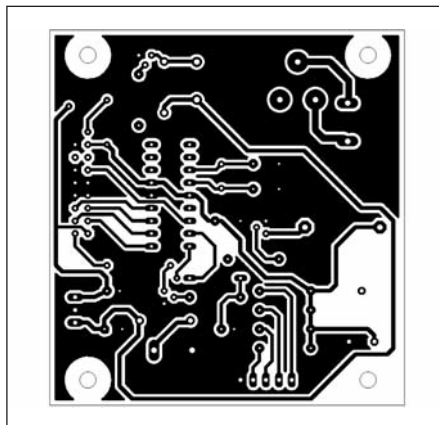



Figura 4: circuito stampato in scala 1:1 (lato saldature).

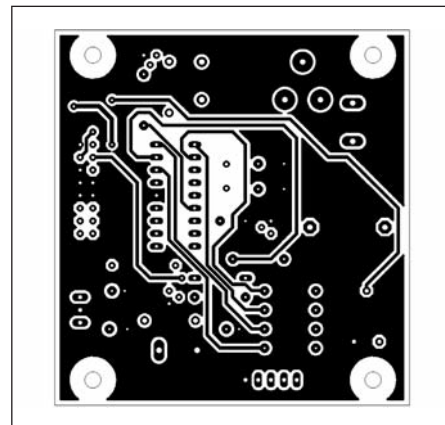



Figura 5: circuito stampato in scala 1:1 (lato componenti).

svolge tutte le funzioni. Alla porta RB è connesso il display LCD tramite il connettore SV1 (RB4, RB3, RB2, RB1 e RB0). Il display LCD, è stato collegato in modalità dati a 4 bit utilizzando solo le linee DB4, DB5, DB6, DB7 e E (Enable per l'attivazione).

Le linee DB0, DB1, DB2 e DB3 non sono utilizzate e sono collegate a massa, così come anche la linea R/W. In questo modo è selezionata la modalità di funzionamento di sola scrittura.

La retroilluminazione è gestita dal PIC, essa è accesa ogni volta che viene premuto un tasto e alla fine di un'esposizio-

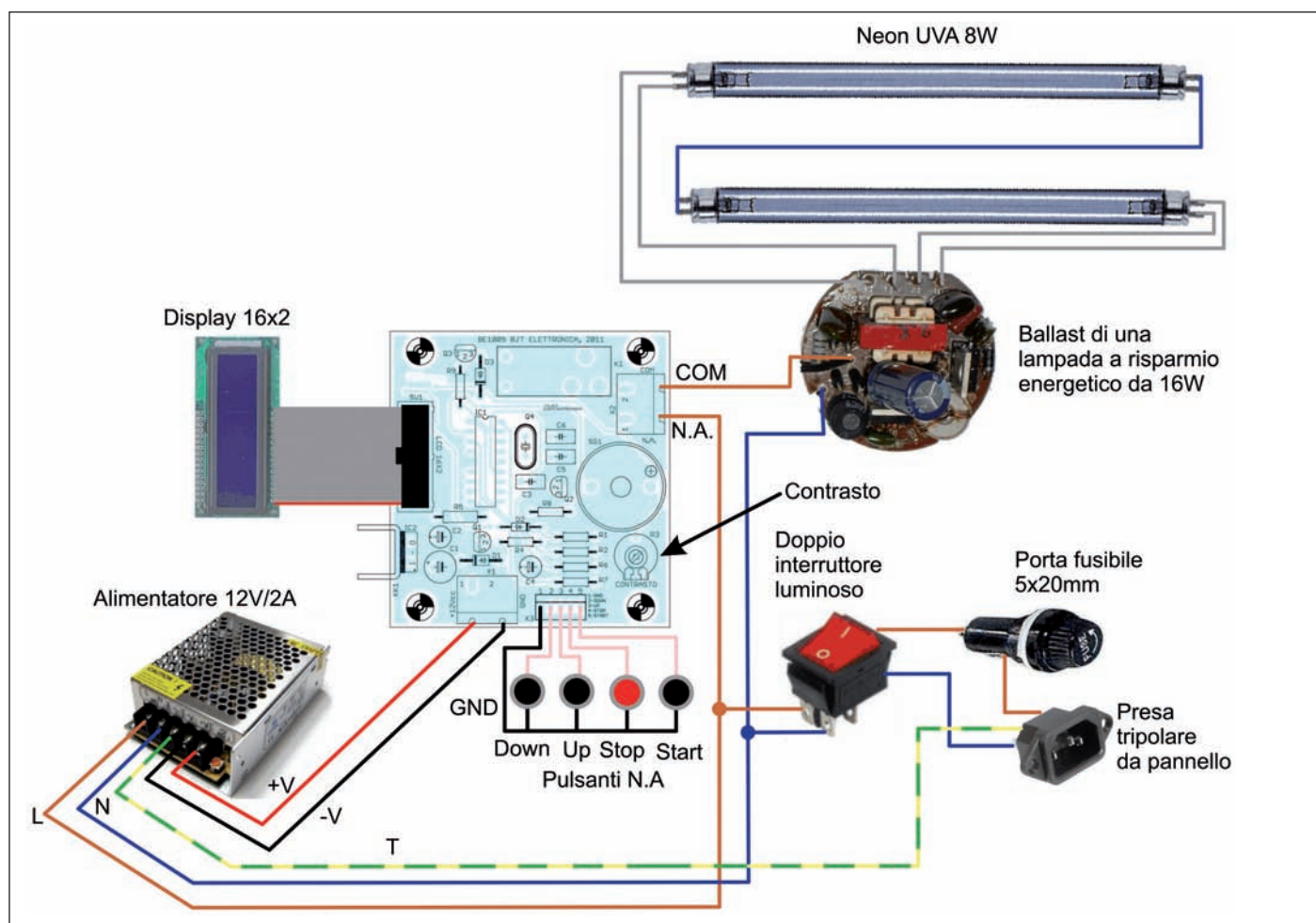



Figura 6: schema di collegamento.

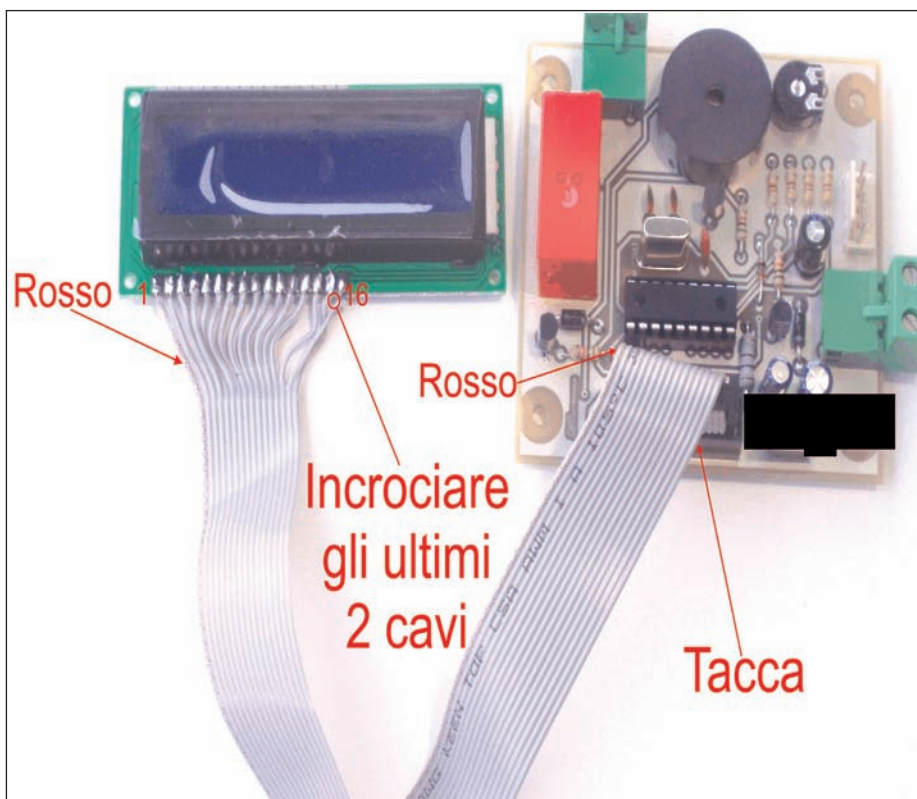



Figura 7: collegamento del Display LCD 16x2.

Figura 8: schermata d'avvio.

Figura 9: schermata impostazioni.

Figura 10: schermata di fine esposizione.

LISTA COMPONENTI

C1	100 uF, 25 V elettrolitico
C2	10 uF, 25 V elettrolitico
C3	100 nF poliestere
C4	22 uF, 25 V elettrolitico
C5-C6	47 pF ceramico
D1	1N4007 diodo
D2	1N4148 diodo
D3	1N4007 diodo
IC1	PIC16F628A
IC2	L7805CV
K1	Relè 1 sc.
Q1-Q2-Q3	BC337 transistor
Q4	Quarzo 4 MHz
R1-R2	4,7 KΩ, 1,/4 W
R3	4,7 KΩ, trimmer orizz.
R4	10 KΩ, 1/4 W
R5	100 Ω, 1/2 W
R6-R7	4,7 KΩ, 1/4 W
R8-R9	10 KΩ, 1/4 W
SG1	Buzzer con elettr. p. 15 mm
SV1	IDC 16 poli
X1-X2	Mors. 2 poli 7,2 mm
X3 Strip	5 poli
N.1	Zoccolo DIL 18
25 cm	Cavo piatto 16 poli
N.1	IDC 16 poli a crimpare
N.1	Display 16X2 blu

ne in modalità lampeggiante. Alla linea RB5 è collegato il transistor Q1, in configurazione d'interruttore elettronico, che consente di attivare la retroilluminazione del display LCD.
Il condensatore C4 permette lo spegnimento soft cioè lo spegnimento graduale della retroilluminazione. D2 si oppone alla corrente di scarica del condensatore C4 evitando il danneggiamento della porta RB5.
Alla porta RB7 fa capo un altro transistor (Q3), in configurazione d'interruttore elettronico che permette di pilotare il relè K1, il quale, a sua volta, consente di attivare l'unità delle lampade UVA.
Il buzzer con elettronica bordo SG1 è pilotato dal transistor Q2 e possiede le tre funzioni di segnalare la fine della foto incisione, quella di tono tasti, e inoltre di segnalare il superamento delle due soglie 1 e 3600 secondi, funzione detta di Error beep.
I resistori R1, R2, R6, e R7 collegati a +5 V costituiscono un pull up alle linee d'ingresso RA0, RA1, RA2, RA4 e consentono di commutare fra due livelli logici 0 e 1, i pulsanti esterni UP, DOWN, START, STOP.
Il diodo D1 protegge da inversioni di polarità accidentali. Il regolatore di tensione IC2 stabilizza la tensione d'ingresso su 5 V, tensione richiesta per alimentare il PIC, mentre la tensione necessaria per alimentare il relè K1 è presa direttamente dall'alimentatore a 12 V stabilizzato esterno.
Il trimmer R3 è usato per regolare il contrasto del display LCD.

REALIZZAZIONE PRATICA

Per la realizzazione di questo progetto **(figura 1)**, si devono utilizzare i master di **figura 4** e **5** per il circuito stampato e i componenti elencati nella **lista componenti**. Il circuito stampato deve essere realizzato con fori metallizzati.
Una volta realizzato il circuito stampato potete iniziare a montare tutti i componenti, disponendoli come **figura 3**.
Si raccomanda di montare per prima tutti componenti a basso profilo, resistenze, condensatori e lo zoccolo, passando ai morsetti e infine al relè.

Protocol Simulator Board

Problemi per analizzare i segnali BUS presenti nel proprio progetto?
Zeroplus viene in aiuto con questa scheda per l'analisi dei protocolli!
Durante lo sviluppo o la decodifica dei segnali, la scheda "PSB" aiuta a visualizzare i pacchetti e ridurre così i tempi di sviluppo migliorando l'efficienza. Lavorando con il Logic Analyzer, si possono valutare le relazioni della base di tempo e dei pacchetti migliorando il progetto.
Ideale sia per i progettisti professionisti che per la didattica.

CODICE MIP 2828949

PC System	Memory	Digital Audio	IC Interface	Optoelectronics	Wireless	Other
Compact Flash 4.1	1-Wire, 3-Wire	AC97	HPI	7-SEGMENT LED	Differential Manchester	BDM, DS 1302
eMMC	I2C	HD Audio	JTAG 2.0	CCIR656	DigRF	DS18B20
FWH	I2C(EEPROM 24L)	HDMI CEC	MCU-51 DECODE	COME IMAGE	ISO7816 UART	DSA Interface
Low Pin Count	I2C(EEPROM 24L	I2S	MICROWIIRE	DALI Interface	KEELOQ Code Hopping	GPIB, HART
LPC-SERIRQ	CS6/24LCS62)	PCM	SIGNIA 6210	DM114/DM115	MANCHESTER	ModBus
LPT, PCI, PECI	MICROWIRE(EEP	S/PDIF	SLE4442	DMX512	MII	MODIFIED SPI
PS/2	ROM 93C)	STBus	SSI Interface	LCD12864	MIL-STD-1553	OPENTHERM 2.2
DS2.0/DSIO, SVID	SPI, SPI PLUS		ST669	LCD1602	MILLER	PSB Interface
UART(RS232C	SPI(EEPROM AT25F)			LG4572	MODIFIED MILLER	PROFIBUS
/422/485)	SAMSUNG			S2Cwire/AS2Cwire	SWP	SHT11, YK-5
USB 1.1	K9(NAND flash)			SCCB	WIEGAND	1-Wire(Advanced)
USB 2.0	UNI/O				WWV/WWVH/WWVB	

Infrared rays	Basic Logic Application	Power	Automotive
IRDS	ARITHMETICAL LOGIC	BMS, HDQ	CAN 2.0B
NEC PD6122	DIGITAL LOGIC	PMBus 1.1	DSI Bus
Philips RC-5	JK FLIP-FLOP	SDQ	FlexRay 2.1A
Philips RC-6	UP DOWN COUNTER	SMBus 2.0	LIN 2.1
PT2262/PT2272			MVB
			WTB

LAP-C Series LAP-B Series

E' un prodotto distribuito da

Per maggiori informazioni www.elettroshop.com oppure chiamare lo 02/66504794

Figura 11: contenitore del bromografo.

Si raccomanda di utilizzare un saldatore da 30 W a punta fine con l'apposita spugnetta ben inumidita.
Dopo avere verificato che tutto è saldato correttamente, bisogna scaricare il file eseguibile del PIC dal sito fare elettronica e programmarlo tramite IC PROG.
In seguito si può procedere a collegare il display LCD al connettore a perforazione rispettando la tacca del connettore ed eseguendo le connessioni di **figura 7**. È importante incrociare gli ultimi due cavetti, altrimenti la retroilluminazione non si accenderà.
In **figura 6** è illustrato lo schema pratico. Prima di eseguire il montaggio, oltre alla scheda con display occorrono:

- 1 alimentatore switching da 12 V, 2 A o superiore;
- 3 pulsanti da pannello del tipo normalmente aperto di colore nero;
- 1 pulsante da pannello del tipo normalmente aperto di colore rosso;
- 1 doppio interruttore luminoso da pannello;
- 1 portafusibile da pannello 5x20 mm;
- 1 presa tripolare da pannello;
- 1 fusibile 5x20mm da 1 A;
- 1 cordone di alimentazione per presa tripolare;
- 1 ballast prelevato da una lampada a risparmio energetico da 16 W;
- 2 lampade uva da 8 W;
- 4 contatti per neon da 8 W;
- vari cavi di collegamento.

Eseguite tutti i collegamenti di **figura 6**. Il ballast va smontato da una lampada a risparmio energetico di basso costo e in seguito collegato come illustrato in **figura 6**.
L'utilizzo del ballast, o reattore elettronico, diminuisce il tempo di accensione delle lampade e ne prolunga la durata.
Realizzate il contenitore del bromografo più adatto alle vostre esigenze e al contenimento delle lampade UVA e della scheda. In **figura 11** è visibile un esempio

di costruzione. Sul sito di Fare Elettronica potete scaricare la maschera serigrafata del bromografo (vedere **figura 11**).
Eseguite il collaudo se il display non visualizza nulla, regolate il contrasto tramite trimmer R3 sulla scheda.
All'azionamento del doppio interruttore viene fornita alimentazione della scheda e sul display viene visualizzato il messaggio di **figura 8** accompagnato dal suono di beep**.** Tutto questo avviene per 5 secondi; in seguito viene mostrata la schermata principale (**figura 9**).
Sulla riga TEMP. ESP (vedere f**igura 9**) si imposta il tempo di esposizione tramite i pulsanti UP (incrementa) e DOWN (decrementa). Il tempo di esposizione varia in secondi. Per cancellare il tempo di esposizione premere a lungo il pulsante STOP.

Una volta impostato, il tempo di esposizione verrà salvato in memoria e anche se il dispositivo viene spento il valore rimarrà in memoria fino quando sarà cancellato o modificato.

Per avviare l'esposizione ai raggi UVA premere START sulla riga TIMER (**figura 9**); in seguito verrà visualizzato l'avanzamento dei secondi fino ad arrivare al tempo impostato. A quel punto si visualizzerà il messaggio di **figura 10** accompagnato nuovamente dal beep e dalla retroilluminazione lampeggiante. Ciò indica che il tempo di esposizione è finito si può premere il pulsante STOP e ritornare alla condizione precedente.
Durante l'avvio di un'esposizione se si desidera fermare l'esposizione è sufficiente premere STOP. Il TIMER verrà azzerato. La retroilluminazione è attivata da qualsiasi pulsate premuto.
È importante che durante l'avvio dell'esposizione venga premuto START per accendere la retroilluminazione. Usare il pulsante STOP per accendere la retroilluminazione solo quando non c'è un'esposizione in corso.

CONCLUSIONI

Nella costruzione del contenitore del bromografo occorre tenere distanti le lampade dal PCB di almeno 5 cm. Il tempo di esposizione consigliato per questa distanza e di 185 secondi ma è opportuno effettuare qualche prova prima di operare la fotoincisione di un circuito, così da ottenere il valore esatto per il vostro fotoresist.
Il firmware e tutti file per necessari alla realizzazione di questo progetto sono disponibili per il download dal sito di Fare Elettronica.
Al seguente link è possibile vedere il temporizzatore in funzione:
http://www.youtube.com/watch?v=ZFMtrJJ1qDc ❑

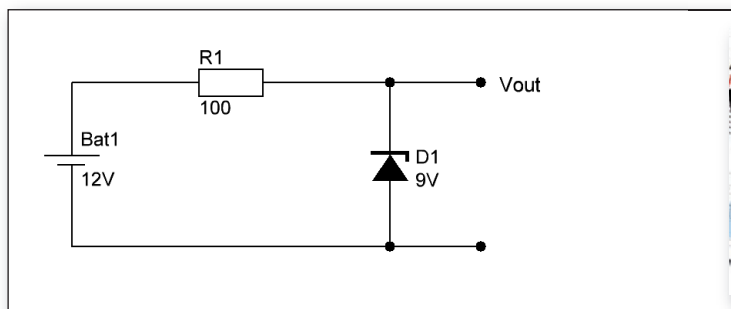
CODICE MIP 2828836

elettro quiz

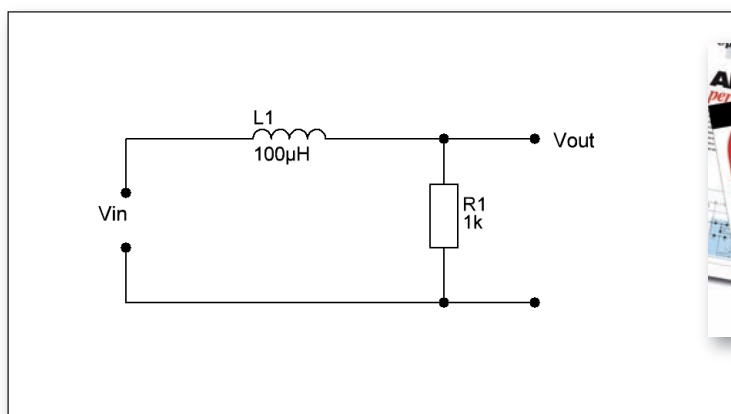
328

Le risposte ai quiz "facile" e "difficile" vanno inviate esclusivamente compilando il modulo su www.farelettronica.com/eq specificando la parola chiave ***"Tesla"****.*

Le risposte e i vincitori (previa autorizzazione) sono pubblicati alla pagina www.farelettronica.com/eq a partire dal 15 del mese successivo alla pubblicazione sulla rivista.

A tutti i partecipanti verrà assegnato un buono sconto del 10% (validità 3 mesi dalla data di assegnazione) utilizzabile per un prossimo acquisto su www.ieshop.it

abbonamento al club di Fare Elettronica

facile

Dato il circuito di figura, in cui il diodo può essere considerato ideale, determinare la tensione di uscita e la corrente nella resistenza R1. Che cosa succede se all'uscita si applica un carico di 1 KOhm?

Abbonamento a Fare Elettronica

difficile

Con riferimento al circuito di figura, quali delle seguenti affermazioni sono vere?

a) *Il circuito è un filtro passa basso*

b) *La componente continua di Vin viene bloccata*

c) *Applicando un carico all'uscita varia la frequenza di taglio*

d) *Il circuito taglia i picchi negativi di Vin*

di ANTONIO GIANNICO

PC EMBEDDED: il progetto RASPBERRY PI

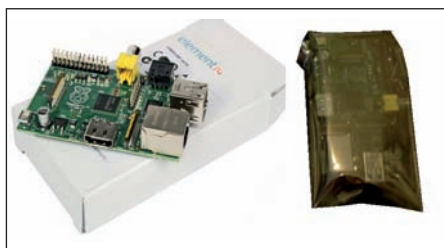
(parte seconda)

In questa puntata getteremo le basi per l'impiego pratico di Raspberry Pi. Porremo, per questo, l'hardware del Raspberry sotto la lente di ingrandimento, ne individueremo i componenti e ne analizzeremo le caratteristiche tecniche. Individueremo quindi gli accessori non in dotazione con la board ma necessari all'allestimento della postazione di lavoro. Concluderemo con un doveroso cenno a che cosa è possibile fare con Raspberry e ai progetti concorrenti attualmente sul mercato.

Nella prima puntata della serie di articoli dedicati a Raspberry Pi si è discusso di come, in un mondo in continuo cambiamento, il personal computer possa divenire pocket a tutti gli effetti e di come il Raspberry Pi altro non sia che l'ultimo e più entusiasmante esempio di questa evoluzione. Questa seconda puntata sarà dedicata soprattutto all'hardware della board, coscienti di quanto un'analisi di questo tipo sia fondamentale nel raggiungimento dei seguenti obiettivi:

1. comprendere appieno le potenzialità, ed eventualmente i limiti, di Raspberry Pi;
2. conoscere il layout della board, i chip presenti a bordo e in che modo essa possa essere interfacciata verso l'esterno (connettori vari e porte di I/O);
3. individuare i componenti accessori non presenti a corredo ma necessari affinché la board possa essere compiutamente utilizzata in tutte le sue potenzialità come PC embedded, quali sono le loro caratteristiche, come reperirli e a quali costi.

Figura 1: Raspberry Pi è confezionato in una busta antistatica racchiusa a sua volta in una scatola di contenimento. Occorre, come con tutte le schede elettroniche, maneggiare con cura la board al fine di evitare danneggiamenti dovuti all'elettricità statica.

Il raggiungimento di questi obiettivi ci consentirà di acquisire basi solide e preparare il terreno alle puntate successive che saranno incentrate su aspetti di carattere maggiormente operativo, in linea con lo stile di chi predilige sperimentare per imparare.
Le conoscenze che acquisiremo in questa puntata si riveleranno utili nel prosieguo, soprattutto perché ci consentiranno di allestire senza dubbi o incertezze una postazione di lavoro Raspberry Pi completa e funzionale in tutte le sue parti. Solo dopo aver allestito compiutamente una tale postazione sarà infatti possibile passare alla fase operativa del nostro percorso ovvero alla preparazione di una SD di boot del sistema operativo e all'impiego effettivo della postazione come PC embedded, su cui far girare sia applicativi "preconfezionati" sia applicazioni nuove sviluppate ad hoc dall'utente, basate sulle potenzialità hardware e software di Raspberry Pi e improntate alla filosofia open source tipica di Linux.
Per chi abbia acquistato o sia in procinto di acquistare Raspberry Pi, ricordiamo che la scheda è confezionata in una busta antistatica racchiusa a sua volta in una scatola di contenimento **(figura 1)**. Occorre, come con tutte le schede elettroniche, maneggiare con cura la board al fine di evitare danneggiamenti dovuti all'elettricità statica.
Prima di proseguire, entrando nel vivo degli aspetti hardware di Raspberry Pi e degli accessori necessari all'allestimento della nostra postazione, è opportuno ricordare brevemente alcuni aspetti del progetto Raspberry Pi, in buona parte già discussi nel precedente articolo e che ci aiutano a comprendere la filosofia del progetto che sta dietro a questo "gioiellino".
L'idea che, nelle intenzioni della fondazione Raspberry Pi, sta alla base del progetto è quella di realizzare uno strumento potente ma economico per l'insegnamento di base dell'informatica e della programmazione. Il Raspberry Pi sta divenendo in effetti tutto questo e inoltre rappresenta una grande opportunità per appassionati programmatori, in considerazione da un lato della qualità e delle potenzialità del prodotto e dell'altro del ridotto costo di acquisto. Considerazioni, queste, che lo pongono sul mercato come l'SBC (Single-Board Computer) del momento. Del resto, nello stesso spazio occupato da una carta di credito, Raspberry Pi mette a disposizione una CPU ARM da 700 MHz con GPU capace di riprodurre video a 1080p e in grado di interfacciarsi a diversi dispositivi di I/O, il tutto a un costo di soli 30 euro circa.
Qualche altra doverosa riflessione: oggi è diffusa la falsa convinzione che saper

usare software come Excel, Word, Autocad e tanti altri applicativi presenti sul mercato sia sinonimo di saper usare un computer. Questo è vero solo in parte. Del resto la tecnologia è affascinante anche per chi non ha una visione tecnica delle cose perché riesce a mostrare il suo lato applicativo sotto forme sempre più immediate e soprattutto intuitive, nascondendo di fatto all'utente finale la propria reale complessità. Questa tendenza ha portato una fascia di persone sempre più ampia della nostra società all'uso massiccio di applicazioni software "preconfezionate" in qualunque ambito e in qualunque attività, sebbene la maggior parte degli utenti, di fatto, non abbia mai programmato un computer o modificato un programma preesistente.
Raspberry Pi, grazie alle sue potenzialità ed al suo ridottissimo costo è, anche per questo, un salto in avanti notevole nella didattica e nell'approccio alla programmazione e al mondo dei computer e rappresenta, per certi aspetti, un ritorno a una filosofia in parte persa nel tempo, quella improntata alla conoscenza della macchina e a alla capacità di interagire quindi con essa a livello più profondo. La fondazione che ha progettato e realizzato Raspberry Pi ritiene, del resto, che molti adolescenti siano oggi utenti passivi del computer e che essi non abbiano, di fatto, alcuna concreta conoscenza del suo reale funzionamento. Raspberry Pi nasce quindi anche con l'intenzione di offrire, soprattutto a giovani studenti, uno strumento per capire come funziona e come si programma un computer.
Questo modo di vedere le cose ad alcuni potrebbe apparire in contraddizione con l'evoluzione tecnologica cui assistiamo. In realtà non lo è affatto. Se è vero che creare programmi ex-novo era una volta una vera e propria necessità, divenuta via via inutile agli occhi di molti che vedono oggi l'applicazione come qualcosa di "preconfezionato", è anche vero che in ambiti quali scuole, facoltà universitarie, enti scientifici e, a volte, anche lavorativi è oggi opportuno più che mai possedere una certa famigliarità con la programmazione e più in generale con la macchina. La filosofia del Raspberry Pi sembra in tutto e per tutto incarnare questo spirito e le parole di Eric Schmidt, presidente di Google sembrano enfatizzarlo in tutta la sua semplicità: "Non tutti i bambini diverranno programmatori, ma non dovrebbe essere negata la possibilità a quelli che ne hanno l'attitudine".
In questo scenario, l'aspetto più eclatante che colpisce al primo impatto è quindi che la fondazione Raspberry Pi sia riuscita a creare il mini-pc che rappresenta, probabilmente, in questo momento, il miglior connubio tra tre principali caratteristiche: costare pochissimo, essere piccolo ed essere potente tanto da risultare adeguato allo svolgimento delle attività di programmazione di un programmatore medio, e persino sostituire un PC standard.

RASPBERRY PI MODEL A E MODEL B

Passiamo ora alla presentazione del Raspberry Pi per quanto concerne gli aspetti hardware. Il Raspberry Pi è un SBC (Single-Board Computer, ovvero un calcolatore implementato su una sola scheda elettronica) sviluppato nel Regno Unito da Raspberry Pi Foundation e progettato per costituire una piattaforma di programmazione basata su kernel Linux o RISC OS. Nelle intenzioni di Raspberry Foundation sono previste due diverse configurazioni hardware di Raspberry Pi, che si presentano entrambe come schede nude, note rispettivamente come Model A e Model B. Entrambe condividono lo stesso equipaggiamento di base e la differenza di costo dovrebbe aggirarsi intorno a 10 euro. In sintesi, mentre il Model B, acquistabile già da alcuni mesi, è dotato di due porte USB e una porta Ethernet LAN 10/100, il Model A, non ancora in produzione al momento della stesura di questo articolo, è dotato di una sola porta USB e non presenta porta Ethernet. Il resto della configurazione è praticamente identica ed è progettata intorno allo stesso processore Broadcom BCM2835 con 256 MB di RAM, CPU ARM1176JZF-S a 700 MHz, acceleratore grafico VideoCore IV a 250 MHZ e decoder hardware in grado di gestire il video Full-HD a 1080p. La dotazione audio comprende un'uscita stereo su jack audio da 3,5 mm per cuffie o altoparlanti amplificati esterni mentre quella video comprende un'uscita video-composito sul classico connettore RCA giallo. Equipaggia la board, sempre in entrambi i modelli, l'uscita audio-video digitale HDMI. Completa in fine la dotazione hardware una porta programmabile GPIO (General Purpose Input Output), la porta DSI (Display Serial Interconnect) per la connessione di TFT touch-screen e la porta CSI (Camera Serial Inerconnect) per la connessione di una eventuale camera digitale. Per quanto riguarda invece l'ingresso audio una piccola nota: non è disponibile ingresso microfonico per cui è necessario disporre di un microfono USB se si vuole acquisire l'audio. Completa la configurazione hardware della board, lo slot per schede SD, che funge tra l'altro da supporto di boot del sistema operativo. La dotazione hardware appena presentata fa di Raspberry Pi un Single Board PC essenziale ma praticamente completo di tutto.
Ciò che non è integrato in maniera nativa sulla board può in ogni caso essere aggiunto dall'esterno compreso l'eventuale connessione Wi-Fi che può essere aggiunta attraverso un modulo WiFi su USB. Le caratteristiche hardware complete sono riassunte per comodità in **tabella 1**, sia per il Modello A sia per il Modello B. Al riguardo è opportuno sottolineare che al momento della stesura del presente articolo il solo modello B di Raspberry Pi è in produzione (distribuito ufficialmente rispettivamente da RS Components **[4]** e da Farnell **[5]**).

L'HARDWARE SOTTO LA LENTE DI INGRANDIMENTO

Il processore del Raspberry Pi (Broadcom BCM2835) è in realtà un vero e proprio SoC (System On Chip) poiché integra in un unico chip, CPU (Central Processing Unit), GPU (Graphic Processing Unit), DSP (Digital Signal Processing) e SDRAM, oltre alle interfacce di I/O precedente riassunte in **tabella 1**. Queste ultime sono rese disponibili, ovvero accessibili, su altrettanti connettori, la maggior parte dei quali di tipo standard, per l'interfacciamento di dispositivi e periferiche esterne, alcune delle quali strettamente necessarie, altre necessarie o meno a seconda dell'impiego che si fa della scheda.
Analogamente a un qualunque PC è infatti necessario che Raspberry Pi venga dotato almeno di una tastiera, un monitor e un alimentatore, da acquistare separatamente in quanto considerati accessori non forniti in dotazione.

TABELLA 1: CARATTERISTICHE TECNICHE DI RASPBERRY PI.

	MODEL A	MODEL B
Prezzo di offerta:	-	33 euro [5]
SoC (System on a Chip)	Broadcom BCM2835 (CPU + GPU + DSP + SDRAM)	
Produttore	Raspberry Pi Foundation (Regno Unito)	
Classe di computer	single-board	
Tastiera incorporata	NO	
Display incorporato	NO	
Unità ottica	NO	
commercializzazione	Prima metà del 2012	-
CPU	700 MHz ARM1176JZF-S core (famiglia ARM11)	
GPU	Broadcom VideoCore IV,OpenGL ES 2.0, 1080p30 H.264/MPEG-4 AVC high-profile decoder	
Scheda video	Broadcom VideoCore IV	
Memory (SDRAM)	256 Megabyte (condivisa con la GPU)	
Porte USB 2.0	1	2 (hub USB integrato)
Output video	Connettore RCA per il video composito, HDMI	
Output audio	3,5 mm jack stereo, HDMI	
Memoria	SD / MMC / SDIO	
Collegamenti di rete	Nessuno	Ethernet 10/100 (RJ-45)
Periferiche di basso livello	Header 2x13 pin GPIO, SPI, I²C, UART, +3,3 V, +5 V	
Real-time clock	No	
Potenza assorbita:	500 mA, (2,5 watt)	700 mA, (3,5 watt)
Alimentazione	5 volt via MicroUSB o GPIO header	
Dimensioni fisiche e peso	85,60 mm × 53,98 mm x 17 mm; peso: 45 g (esclusa la sporgenza dovuta alla SD e ai connettori);	
Sistemi operativi supportati	Debian GNU/Linux, Raspbian, Fedora, Arch Linux,Gentoo	
Sistemi operativi non supportati:	RISC OS (shared source)	
linguaggio di programmazione consigliato	Python	
Sito Web di riferimento	http://www.raspberrypi.org	

La CPU integrata nel SoC Broadcom BCM2835 è un ARM1176JZF-S con frequenza di clock di 700 MHz, la GPU è invece una VideoCore IV, la memoria integrata è pari a 256 MB RAM, condivisa tra CPU e GPU. L'architettura, come avviene anche per altri PC on Board più costosi, non prevede l'impiego né di dischi rigidi per il SO né di unità a stato solido bensì di una scheda di memoria di tipo SD utile sia per il boot sia come memoria non volatile.

La prima differenza sostanziale che il Raspberry Pi mostra, alla pari di altri PC embedded, rispetto alla struttura di un PC standard è quindi l'assenza di un hard-disk. Del resto, la connessione di un hard disk esterno per l'installazione del sistema operativo o di software applicativo, in linea di massima contrasterebbe con la logica di rendere il sistema minimale nella sua composizione e quindi il più compatto ed economico possibile.

L'hard disk è per questo sostituito dalla presenza di uno slot per SD (Flash Memory Card) dello stesso tipo di quelle utilizzate comunemente per fotocamere e videocamere e che viene vista dal processore come se si trattasse di un hard-disk. Il sistema operativo viene quindi caricato in RAM proprio dalla SD card all'avvio, a differenza di quanto accade in un PC standard che normalmente carica il sistema operativo dall'hard disk durante la fase di boot.

Continuando la nostra analisi, va detto che può sembrare strano il fatto che sia stata prevista sulla board l'uscita video-composito, tecnologicamente obsoleta. Tale scelta deriva dalla filosofia che è all'origine della nascita del progetto Raspberry cioè dall'intenzione di rendere disponibile al maggior numero di persone possibile, specialmente a chi ha modeste possibilità economiche, la possibilità di usare un computer, unitamente al fatto che in molti parti del mondo sono ancora diffusi i vecchi TV con ingresso composito.

Il costo ridotto non è in ogni caso l'unica nota attraente di Raspberry Pi. Attrae la filosofia open source che ha contribuito al successo di altre board di questo tipo (Arduino in particolare, giusto per citarne una piuttosto recente, anche se le due co-

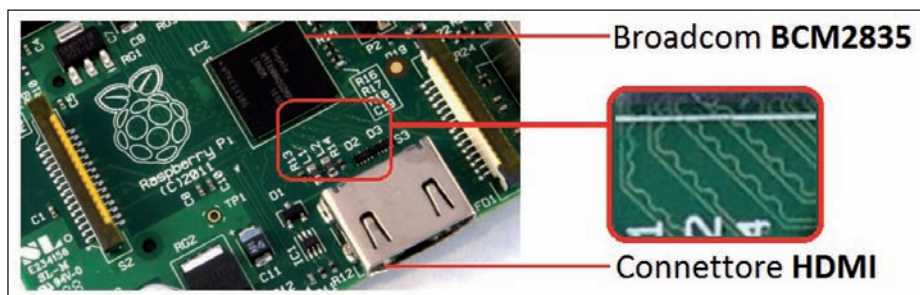



Figura 2: dettaglio delle piste di interconnessione tra processore PCM2835 e connettore audio-video HDMI. La forma ondulata delle piste è studiata per consentire di ridurre interferenza e i ritardi di segnale. La soluzione è adottata in questo caso essendo, quelli trasportati, segnali video digitali.

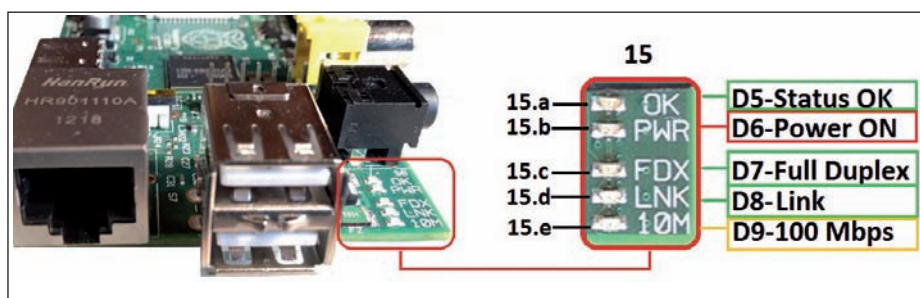



Figura 3: LED di stato del Raspberry Pi D5, D6, D7, D8, D9 (vedi anche tabella 4).

se, come vedremo, non sono tecnicamente paragonabili) e non è da meno la potenza di calcolo assicurata dal processore ARM11+GPU a 32 bit, in grado di far girare, senza problemi, le più disparate applicazioni Linux. La scelta progettuale di impiegare un ARM 11 consente infatti di perseguire l'ottimizzazione di costi da un lato e di performance dall'altro. Tanto per intenderci, si tratta di un processore ARM tipicamente impiegato su smartphone e tablet, nota sufficiente anche ai meno esperti a comprenderne, o per lo meno intuirne, le potenzialità.
La GPU Videocore integrata consente prestazioni multimediali, in termini di frame-rate e risoluzione, di tutto rispetto.
Ovviamente, essendo la RAM integrata nel chip del processore, non può essere ampliata. A tale proposito, va osservato che 256 MB potrebbero sembrare pochi, soprattutto se paragonati con le RAM dei comuni PC da ufficio, tuttavia viene in aiuto il fatto che i sistemi operativi GNU/Linux in combinazione con i processori ARM sono in grado di ottimizzare al meglio l'uso delle risorse di sistema riducendo la quantità di RAM di fatto richiesta senza che questo vada a scapito di un funzionamento fluido e affidabile della macchina.
Evitiamo di entrare nei meandri del processore in maniera dettagliata limitandoci a osservare, oltre a quanto già detto, che il processore Broadcom BCM2835 integra all'interno le seguenti periferiche e risorse hardware:

- timer;
- interrupt controller;
- GPIO (General Purpose Input Output; una parte dei quali utilizzata per esempio per il pilotaggio della SD Card, una parte portata in uscita sul connettore GPIO serigrafato come P1 e una parte ancora inutilizzata);
- USB (una sola porta USB, ampliata a due sulla board raspberry attraverso un hub integrato on-board realizzato grazie al chip LAN 9512 e portate sul connettore USB a due livelli S7, visibile in **figura 3**);
- PCM/I2S;
- DMA controller;
- I2C master;
- I2C / SPI slave
- SPI0, SPI1, SPI2;
- PWM;
- UART0, UART1;

Interfacce di sviluppo (vedi **tabella 3** e **figura 4**):

- CSI (resa disponibile sul connettore S5);
- JTAG (resa disponibile sul connettore P2);
- GPIO (General Purpose Input Output, pin resi disponibili sul connettore P1);
- DSI (reso disponibile sul connettore S2);

La porta Ethernet 10/100 della board è invece resa disponibile attraverso il chip LAN9512, motivo per cui non è supportata la connessione Gigabit.
Per la verifica di quanto esposto e ulteriori dettagli sull'hardware della board si rimanda il lettore all'analisi diretta e più approfondita dello schema elettrico **[10]**.
Alla luce di quanto esposto, possiamo affermare di trovarci di fronte ad un prodotto che rappresenta da un lato l'essenza stessa di un PC Desktop e dall'altro uno strumento che può risultare flessibilissimo come piattaforma per la prototipazione di sistemi embedded, il tutto ad un costo davvero esiguo. Raspberry Pi è in altre parole un mini-PC certamente essenziale ma aperto a tutto.
Una nota importante che potrebbe apparire limitante è il fatto che Raspberry PI non è dotato di real-time clock (RTC). Questo rientra tuttavia nella filosofie di essenzialità ed economicità del progetto. L'integrazione di un RTC necessiterebbe infatti, tra l'altro, di una batteria e di ulteriore spazio sulla board. Si può in ogni caso ovviare a questa carenza attraverso un network time server,

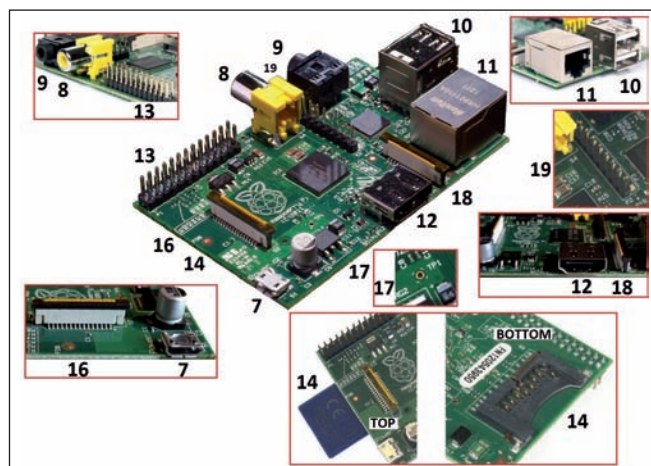



Figura 4: connettori e test point. MicroUSB di alimentazione (7-S1), uscita video HDMI (12-S3), RCA-video-composito (8-S4), uscita audio (9-S6-presa jack 3,5 mm stereo), presa di rete RJ45 (11-P4), doppio connettore USB 2.0 (10-S7), connettore per camera (18-S5), connettore per display touch (16-S2), connettore uscite GPIO (13-P1), connettore JTAG (19-P2), connettore per SD (14-S8).

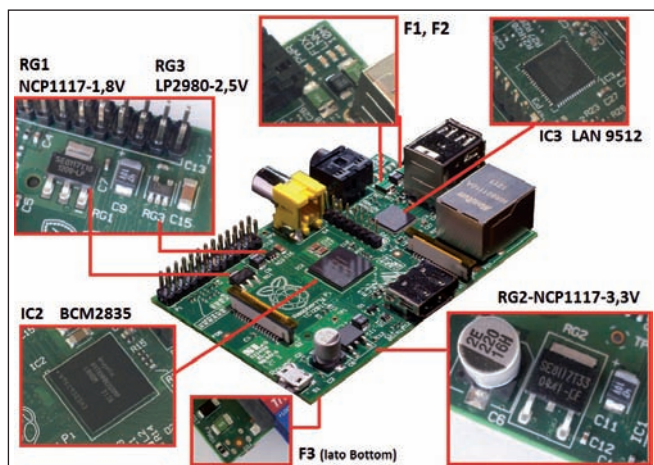



Figura 5: Broadcom BCM2835 (CPU+GPU+DSP+SDRAM), controller LAN (LAN9512), regolatori RG1(NCP1117-1,8V), RG3(LP2980 2,5V), RG2(NCP1117 3,3V); vedere anche figura 7.

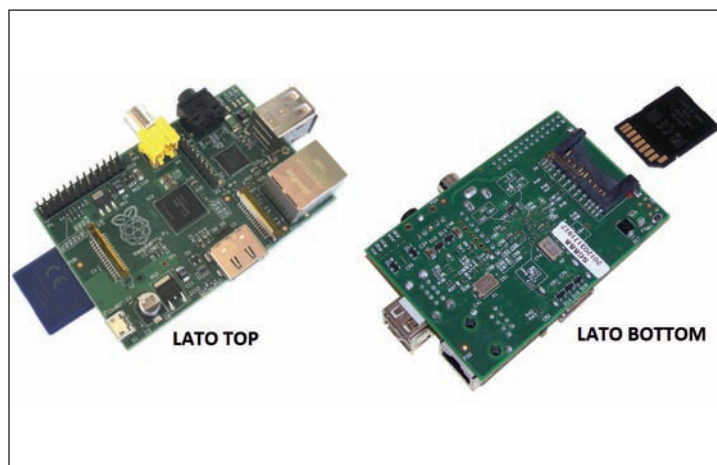



Figura 6: vista lato top e lato bottom del Raspberry Pi (si noti l'alloggiamento per la SD card sul lato bottom).

o più semplicemente inserendo manualmente l'ora allo startup o ancora realizzando l'aggiunta di un RTC, per esempio tramite il chip DS1307 con relativa batteria tampone da interfacciare, mediante l'impiego dei pin GPIO, su interfaccia I²C.

Figura 7: Raspberry Pi - sezione di alimentazione (si notino i tre regolatori per l'ottenimento delle tensioni 3,3 V, 2,5 V e 1,8 V dalla tensione di alimentazione 5 V principale addotta attraverso il connettore microUSB).

LAYOUT E LE PORTE DI I/O

Dopo aver passato sotto la lente di ingrandimento le caratteristiche e le potenzialità hardware del Raspberry Pi, in questo paragrafo intendiamo esaminare, con maggiore attenzione, il layout della board al fine di individuare, sulla stessa, i chip che la compongono, le relative funzionalità e i connettori disseminati sulla sua area per l'interfacciamento delle periferiche esterne. Senza dilungarci eccessivamente, riportiamo sinteticamente, nelle **tabelle 2**, **3** e **4**, l'elenco delle parti componenti la scheda e dei connettori resi disponibili lungo la periferia della stessa e all'interno dell'area della board. Nelle **figure 2**, **3**, **4** e **5** riportiamo la visione di assieme della board e i dettagli delle singole parti, prendendo come riferimento il Modello B di Raspberry Pi.

La scheda dispone di cinque led di stato sul lato del connettore USB (si veda **tabella 4** unitamente a **figura 3**); i primi due (Power ON e Staus OK) indicano, rispettivamente, la presenza dell'alimentazione e la presenza della scheda SD nel suo slot, gli altri tre indicano invece lo stato della rete (Link, Full Duplex e 100 Mbps).

Una piccola nota: il modello A del Raspberry Pi potrebbe apparire fortemente penalizzato dalla mancanza di una porta Ethernet. Questo è vero solo in parte essendo sempre possibile utilizzare un'antenna Wi-Fi tramite la porta USB per navigare senza fili.

In **figura 6** viene mostrato sia il lato TOP sia il lato Bottom della scheda. Come si può osservare, i connettori sono tutti dislocati sul lato TOP in modo da essere facilmente accessibili, escluso lo slot SD, mentre sul lato Bottom trovano dislocazione i due quarzi per le sezioni di oscillazione del processore BCM2835 e dell'interfaccia di rete LAN9512.

SCHEMA ELETTRICO

In questo paragrafo daremo uno sguardo allo schema elettrico della scheda Raspberry Pi. L'hardware, come abbiamo già avuto modo di mostrare, si basa essenzialmente su due chip:

- il processore Broadcom BCM2835, facilmente localizzabile al centro della board;
- il driver LAN9512 che serve a rendere disponibili le due porte USB 2.0 e la porta ethernet 10/100 e che è facilmente localizzabile proprio in prossimità dei connettori USB ed RJ45.

Lo schema elettrico, reso disponibile da Raspberry foundation **[10]**, consta di cinque pagine, ciascuna delle quali riporta una sezione ben precisa dello schema elettri-

TABELLA 2

PARTE	DESCRIZIONE	PARTE DI SCHEMATICO
1	Regolatore di tensione NCP1117 (3,3 V)	RG2
2	Regolatore di tensione LP2980 (2,5 V) per l'uscita video composita (Pin 5)	RG3
3	Regolatore di tensione NCP1117 (1,8V) usato per la SDRAM e la CPU (Pin 2)	RG1
4	Termistori/Fusibili di protezione per limitare la corrente che fluisce all'interno delle periferiche USB (F1-140 mA e F2-140 mA) e la corrente di alimentazione della scheda (F3-1,1 A)	F1, F2, F3
5	256Mb SDRAM interna, processore ARM11 (ARM1176JZF-S) a 700 MHz e GPU VideoCore IV	IC2
6	Chip LAN9512 (offre connessione USB e connessione Ethernet LAN 10/100Mbps via cavo)	IC3

Tabella 2: chip e fusibili di protezione on board (vedi figura 5 e figura 7).

TABELLA 3

PARTE	DESCRIZIONE	PARTE DI SCHEMATICO
7	Micro USB (collegamento Alimentatore micro USB)	S1
8	Uscita video composito RCA (per l'impiego, come monitor, delle TV di vecchia generazione)	S4
9	Uscita audio stereo analogica (connessione a cuffie, ad un amplificatore esterno o a casse acustiche amplificate)	S6
10	Due connettori USB 2.0 (connettore a doppio livello; connessione di tastiera, mouse, webcam, hub USB)	S7
11	RJ45 Ethernet LAN 10/100 (connessione a Internet o a una rete LAN)	P4
12	Uscita audio-video HDMI (connessione delle moderne TV o monitor dotati di ingresso audio-video HDMI)	S3
13	Connettore GPIO (General Purpose Input Output per la connessione di periferiche esterne)	P1
14	SD Card Slot. Alloggiamento per scheda SD contenente il sistema operativo e la collezione di software utente per creare file, navigare in internet, ascoltare musica, scrivere, eseguire lavori di office automation, giocare ecc.	S9
16	DSI (Display Serial Interface) per il collegamento diretto di un opportuno display.	S2
17	Tensione di 5 V (TP1)	TP1
18	CSI (Camera Serial Interface). Interfaccia di connessione seriale videocamera) Consentirà, nel prossimo futuro, la connessione di una telecamera. La GPU potenzialmente consente di gestire acquisire video a 40 MPixel a 1080p e 30 fps	S5
19	JTAG	P2

Tabella 3: connettori e test point (vedi figura 4).

TABELLA 4

PARTE FIG. 3	DESCRIZIONE	PARTE DI SCHEMATICO	COLORE
15.a	Status ok	D5	Verde
15.b	Power On	D6	Rosso
15.c	Full Duplex	D7	Verde
15.d	Link	D8	Verde
15.e	100 Mbps	D9	Giallo

Tabella 4: LED di stato (D5...D9). Vedi Figura 3 per dettagli.

co della board. A questo facciamo esplicito riferimento pur limitandoci a riportare qui esplicitamente la sola sezione di alimentazione **(figura 7)**. Rimandiamo il lettore all'analisi diretta del documento **[10]** contenente lo schematico, direttamente reperibile in rete. Nell'ordine abbiamo:

1. sezione di alimentazione e oscillazione (foglio 1);
2. sezione comprendente l'uscita HDMI, lo slot SD card, il segnale video-composito e la presa jack audio stereo da 3,5 mm. Tutti questi connettori sono connessi tramite gli opportuni componenti passivi di interfacciamento al chip BCM2835, il quale integra le suddette uscite (foglio 2);
3. sezione contenente l'interfacciamento ethernet e USB verso gli omonimi connettori. Tale interfacciamento è reso possibile attraverso il chip LAN9512 che, interconnesso con il processore BCM2835, consente di portare a due il numero di porte USB e di aggiungere alla board una porta ethernet 10/100. I LED di stato della comunicazione ethernet sono pilotati proprio dal chip LAN9512. Si tratta dei LED Full Duplex, Link e 100M (foglio 3).
4. sezione che riporta i connettori JTAG

Figura 9: alimentatore microUSB 6 W (90-264Vin 5 Vdc out, Iout = 1,2 A; costo orientativo: 5-15 euro) [5].

Figura 10: alimentatore 230Vac/5Vdc USB out (costo: 9 euro circa), 1 A, Input Voltage VAC: da 100 V a 240, potenza: 5 W [5].

(connettore 8 poli localizzato nelle vicinanze della porta video-composito), il connettore DSI per la connessione di un eventuale display TFT touch-screen e il connettore CSI per la collegamento di una eventuale telecamera digitale (foglio 4).
La sezione di alimentazione prevede una tensione in ingresso di 5 V (con alimentatore microUSB in grado di assicurare almeno 700 mA di corrente) protetta tramite fusibile/termistore (F1) e una cascata di tre regolatori di tensione:

- NCP1117-3V3 necessario a ottenere 3,3 V;
- LP2980 necessario a ottenere la tensione di 2,5 V;
- NCP1117-1V8 necessario a ottenere la tensione di 1,8 V.

Il processore BCM2835 necessita, infatti, della tensione di 5 V ma anche della tensione di 1,8 V – stabilizzata per la sezione SDRAM, per la sezione HDMI, USB, LAN e per la sezione di oscillazione, – della tensione stabilizzata 3,3 V – per il funzionamento del core, della sezione SD card e GPIO oltre che della sezione USB e di quella LAN, – e, infine, della tensione stabilizzata di 2,5 V per l'alimentazione della sezione DAC del processore. Si rimanda all'analisi diretta dello schema elettrico per maggiori dettagli in merito.

ALLESTIRE UNA POSTAZIONE RASPBERRY: HARDWARE ACCESSORIO E PERIFERICHE

In questo paragrafo analizzeremo l'hardware di cui è necessario disporre per allestire una postazione di lavoro completa intorno all'SBC (Single Board Computer) Raspberry Pi pronta all'uso. Fino a questo momento abbiamo esaminato il Raspberry Pi nel suo equipaggiamento onboard e per quanto attiene le porte di I/O e i connettori disponibili per l'interfacciamento di dispositivi esterni. Se è vero che un PC embedded come il Raspberry Pi può, in alcuni casi, essere utilizzato anche senza alcun hardware esterno, a parte ovviamente un alimentatore (si pensi, per esempio, al controllo automatico di un processo su rete ethernet dove il Raspberry si comporti come una specie di centralina di controllo) è anche vero che nell'impiego più comune, e per noi più naturale, sarà necessario collegare alla board alcune periferiche del tutto analoghe a quelle di un comune PC desktop.

Figura 8: schema generale di connessione del Raspberry Pi alle diverse periferiche necessarie a renderlo pronto all'uso (fonte Farnell [5]: il codice riportato sotto ciascun componente rappresenta il codice prodotto Farnell identificativo per l'eventuale ordine di acquisto) [5].

È importante sottolineare, d'altro canto, che qualunque dispositivo esterno, compreso l'alimentatore microUSB e la SD card contenente il sistema operativo, sono considerati accessori e pertanto non disponibili in dotazione. Chi desidera acquistare la board e impiegarla dovrà pertanto munirsi, a parte, di questi dispositivi, salvo che non ne disponga preventivamente. Per esempio, alcuni potrebbero pensare di utilizzare mouse e tastiera del proprio PC desktop al fine di non far lievitare eccessivamente i costi. Il dispositivo in ogni caso più costoso per l'interfacciamento della board è certamente il monitor, motivo per cui, quasi sicuramente, chi avrà modo di dilettarsi con Raspberry Pi utilizzerà il Monitor del proprio PC oppure il proprio monitor TV.
Fatta questa doverosa premessa, possiamo individuare nel seguente il kit minimale di componenti e accessori necessari per allestire una postazione e cominciare a usare Raspberry Pi:

1. scheda Raspberry Pi nella versione Model a o Model B;
2. SD card contenente un'adeguata distribuzione del sistema operativo Linux necessaria per il boot;
3. tastiera USB;
4. monitor o monitor TV dotato di ingresso video HDMI, DVI, Video-Composito o su presa SCART;
5. alimentatore microUSB 5 V (corrente non inferiore a 700 mA);
6. cavo video e/o adeguato adattatore per la connessione di un TV o monitor con ingresso HDMI, DVI, Video-Composito o SCART all'uscita HDMI del Raspberry o a quella Video-Composito;
7. potrebbe risultare necessario anche un Hub USB powered, soprattutto se si opera con il modello A del Raspberry Pi che, a differenza del modello B, risulta dotato di una sola porta USB.

Tutto questo non esclude, in ogni caso, che nel prossimo futuro non arrivi una sorta di bundle che possa comprendere, oltre alla board, anche accessori quali l'alimentatore Micro USB, mouse e tastiera, eventualmente wireless, memory card SD con distribuzione Linux preinstallata (magari Linux Raspbian, una versione di Linux ottimizzata per l'hardware del Raspberry e basata su Debian), adattatore Wi-Fi, magari webcam da 5 me-

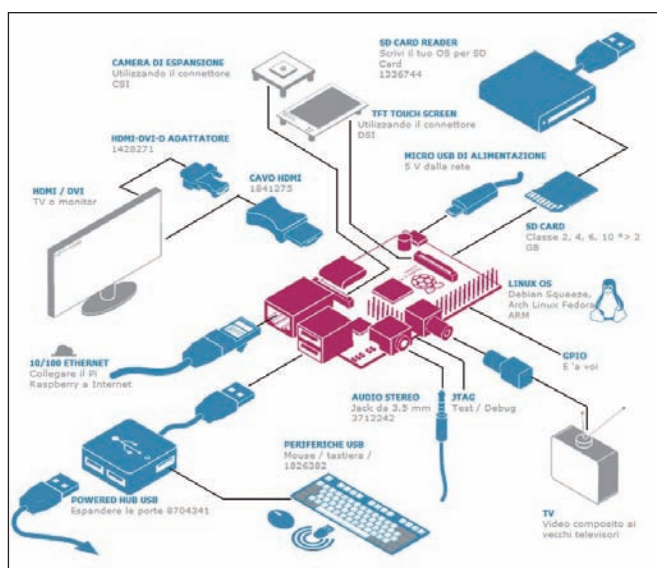



Figura 11: cavo USB A maschio-micro USB maschio (costo: 3 euro circa) [5].

Figura 12: cavo HDMI 19 pin maschio-maschio tipicamente impiagato per il segnale Digital TV; è disponibile in commercio con lunghezze che vanno da mezzo metro a circa 15 metri [5].

Figura 13: mini USB keyboard (costo: 10-14 euro); mini mouse USB (costo: 5 euro circa) [5].

Figura 14: cavo adattatore da USB a 2xPS2 (consente con un solo cavo di connettere mouse PS2 e tastiera PS2 a una porta USB del PC; costo: 30 euro circa) [5].

gapixel. La possibilità di collegare webcam o fotocamere al connettore CSI, tramite un semplice cavo flat, potrebbe rendere infatti il RaspBerry Pi ancor più flessibile poiché aperto ad applicazioni quali quelle di home-automation.

Nel caso in cui si desideri connettere in rete il Raspberry Pi, sarà inoltre necessario dotarsi di un cavo Ethernet per la connessione nel caso di Raspberry modello B oppure di un adattatore USB Wi-Fi nel caso di modello A (ricordiamo, infatti, che il modello B integra il connettore di rete RJ45, mentre il modello A ne è sprovvisto).

Sebbene non fondamentale è inoltre opportuno disporre di un case per il contenimento della scheda, al fine di assicurarle l'adeguata protezione meccanica ed elettrica. La scheda viene infatti venduta "nuda", cioè senza contenitore, allo scopo di ridurre i costi di produzione e rendere disponibile lo "strumento" a una fascia di persone la più ampia possibile. In ogni caso, un package adeguato dovrebbe possibilmente essere trasparente in modo da rendere identificabile ogni singolo componente oltre ai LED di stato di cui la scheda è dotata.

Quanto esposto nel presente paragrafo è efficacemente riassunto in **figura 8**. Lo schema è tratto dal sito web del distributore Farnell **[5]** che, come RS Components **[4]**, è distributore non solo del Raspberry Pi ma anche dell'hardware accessorio. Il codice riportato sotto ciascun componente rappresenta il codice prodotto Farnell identificativo per l'acquisto del medesimo.

Di ciascuno dei componenti ora richiamati o individuati nella suddetta **figura 8** analizzeremo, nel successivo paragrafo, le caratteristiche e i costi, dando indicazioni per il reperimento sul mercato. I cavi accessori e gli adattatori di cui parleremo non devono necessariamente essere tutti disponibili per realizzare la nostra postazione di lavoro; tutto dipende essenzialmente, come avremo modo di comprendere, dal tipo di monitor, mouse e tastiera di cui disponiamo.

Alimentazione

L'alimentazione viene addotta al Raspberry Pi attraverso alimentatore con connettore Micro USB. Come è noto, questo tipo di connettore presenta, secondo un pin-out standard, sia pin di segnale sia pin di alimentazione. Nella fattispecie, vengono utilizzati i soli pin di alimentazione. Si tratta di alimentatori divenuti ormai standard nella telefonia cellulare e pertanto di facile reperibilità presso qualunque negozio di elettronica o di telefonia cellulare, oltre che presso i distributori ufficiali di Raspberry Pi cioè RS Components **[4]** e Farnell **[5]**. La sua reperibilità non pone pertanto alcun problema. L'unico limite cui è necessario porre qualche attenzione è la potenza che l'alimentatore è in grado di erogare (almeno 700 mA a 5 volt). Ricordiamo, a tal proposito, che Farnell e RS Components propongono alcuni di questi alimentatori sul proprio sito come accessori specifici per Raspberry Pi, cosa che dovrebbe consentirvi di dissipare qualunque eventuale dubbio in merito.

L'alimentazione della board potrebbe in ogni caso seguire anche strade alternative. Per esempio, è possibile utilizzare un cavo USB-microUSB e alimentare la board attraverso una porta USB del proprio computer oppure del proprio Hub USB. Il cavo necessario per rendere possibile un tale adattamento è del tipo da USB A "maschio" a USB micro "maschio" **(figura 11)** cioè un tipo di cavo abbastanza diffuso anche come cavo dati per la connessione di piccole apparecchiature portatili ai comuni PC. In realtà questo modo di agire merita una precisazione. Si è infatti detto dell'opportunità di utilizzare un alimentatore in grado di fornire, a 5 V, una corrente di almeno 700 mA. A tale proposito, bisognerebbe ricordare che una singola porta USB di un PC o di un HUB powered può fornire al più 500 mA di corrente. Per questo motivo è comunque opportuno, in questo caso, utilizzare uno speciale cavo USB, normalmente reperibile in commercio, che presenta lato USB a due teste e per questo collegabile a due differenti porte USB contemporaneamente e, dal lato micro USB, il solito connettore microUSB. Poiché lo scopo è semplicemente quello di addurre alimentazione, esso combina la corrente disponibile su ciascuna porta in un'unica alimentazione disponibile per il Raspberry, garantendo fino a 2 x 500 mA = 1000 mA di corrente, ben superiore ai 700 mA richiesti. Nota: Più precisamente, sul documento **[3]** viene dichiarata una corrente massima assorbita di 500-700 mA per il Raspberry Modello A e 700-1200 mA per il Raspberry Modello B.

La potenza assorbita dal solo processore è in realtà legata anche alla frequenza

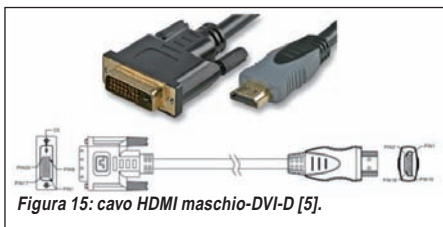



Figura 15: cavo HDMI maschio-DVI-D [5].

Figura 16: adattatore audio-video HDMI Maschio-DVI Femmina [5].

Figura 17: adattatore HDMI Femmina-DVI maschio [5].

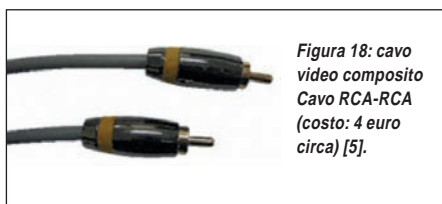



Figura 18: cavo video composito Cavo RCA-RCA (costo: 4 euro circa) [5].

Figura 19: a sinistra la scheda SD 4 GB classe 6, OS Raspberry Pi (Codice RS 763-1030; costo: 13 euro); a destra la scheda SD 4 GB precaricata con Debian 6 Linux OS (Operating Voltage Range: 2.7 V to 3.6 V; Speed Class Rating: 4; Data Rate: 4 Mbps) [4], [5].

Figura 20: USB Card Reader (costo: 6 euro) [5].

di clock dello stesso e cresce proporzionalmente con essa. Per questo motivo è pertanto il caso di ricordare come sia inopportuno addentrarsi in esperimenti che comportino l'overclocking del processore, sebbene, a quanto risulta in rete, qualche "pioniere" sembra avere già avuto modo di eseguire simili esperimenti sul Raspberry Pi. Una tale avvertenza è segnalata anche nel documento **[3]** che recita le seguenti parole: "This product should not be overclocked as this may make certain components very hot". Lo stesso documento consiglia, come è logico, di non chiudere completamente la board in un case privo di finestrature di scambio termico.
Esistono in commercio anche alimentatori come quello in **figura 10** dotato di un'uscita USB lato DC cui è necessario connettere un cavo USB-microUSB per consentire l'alimentazione della board.
Qualche "smanettone", nel senso più nobile del termine, potrà persino pensare di alimentare il Raspberry Pi direttamente dal proprio alimentatore da banco riciclando, e modificando magari, qualche vecchio cavo con terminazione microUSB, oppure potrà costruirsi un circuito con pile AA da 1,5 V in configurazione serie-parallelo per alimentare a batterie il Raspberry. Qualcun altro potrebbe invece pensare di modificare il cavo di alimentazione con terminazione micro-USB inserendovi un diodo in serie e di dotare la board di un'alimentazione a batteria posta in parallelo in tampone con la prima attraverso diodi di disaccoppiamento.
Se invece qualcuno dovesse chiedersi se è possibile alimentare la board tramite Ethernet (PoE) sappia che tale modalità di alimentazione non è contemplata.
Una nota opportuna va invece rivolta a chi possiede un cellulare di ultima generazione. Essendo infatti questi dotati di connettore microUSB, consigliamo di controllarne i dati di targa. Nel caso in cui la corrente che questo è in grado di fornire sia maggiore o uguale a 700 mA, si potrà utilizzarlo senza problemi di sorta per alimentare il piccolo Raspberry, evitando così di dover acquistare un nuovo alimentatore.

Monitor video

Molti lettori si staranno certamente chiedendo come collegare il monitor del loro PC al Raspberry. La maggior parte dei monitor TV attuali è dotata di ingresso video HDMI, mentre la maggior parte dei monitor per PC che non integrano il ricevitore TV presenta ingresso DVI oppure doppio ingresso (DVI e VGA) o ancora il solo ingresso VGA nel caso di modelli più datati.
Il Raspberry Pi dispone di due uscite video:

- HDMI (high definition);
- Video Composito;

È necessario allora comprendere quali possibilità di connessione diretta, o tramite opportuni adattatori, sussistano tra i due dispositivi.
In linea di massima, la presenza della doppia uscita sul Raspberry Pi consente di connettere sia un TV analogico che digitale oppure un monitor con ingresso DVI. Vecchi monitor TV ma anche TV più recenti dispongono inoltre di ingresso video-composito (per intenderci si tratta fisicamente del connettore giallo tondo). Questi possono essere connessi al Raspberry attraverso un comune cavo video-composito con connettori gialli RCA. La maggior parte dei dispositivi TV integra inoltre, ancora oggi, la presa SCART. Anche in questo caso è possibile sfruttare opportuni adattatori da video-composito a SCART per la connessione.
Si tratta di convertitori passivi, poco costosi e reperibili in maniera relativamente semplice in negozi di elettronica consumer oppure presso comuni distributori come Farnell **[5]** o RS **[4]**.
Per quanto riguarda invece i monitor che dispongono di solo ingresso VGA, il loro eventuale impiego è meno agevole poiché il Raspberry Pi non è dotato di uscita VGA (anche in considerazione del fatto che la tecnologia VGA è ormai quasi soppiantata da quella DVI e HDMI). Qualche lettore starà probabilmente pensando all'eventualità di eseguire una doppia conversione, cioè di utilizzare un primo adattatore da HDMI a DVI-D (digital) cui far seguire un successivo adattatore da DVI a VGA, entrambi adattatori o cavi adattatori passivi disponibili in commercio. Purtroppo questa soluzione non funziona essendo l'HDMI puramente digitale e non contenendo il segnale DVI-A (analogue) che sarebbe invece necessario per la conversione a VGA. Ne consegue che la conversione da HDMI o DVI-D a VGA è in linea di massima possibile ma richiede appositi convertitori attivi poco economici.
Nei paragrafi che seguono passeremo in rassegna i diversi cavi e adattatori che potranno risultare utili per la connessione video, e non solo, del Raspberry.

Cavo video HDMI

Nel caso in cui si disponga di un monitor o un monitor TV con ingresso HDMI è necessario utilizzare un cavo HDMI maschio-maschio per la connessione all'uscita video HDMI del Raspberry Pi, disponibile in commercio con lunghezze che vanno da mezzo metro a 15 metri.
Si noti che in realtà si parla impropriamente di cavo video in quanto lo standard HDMI supporta, in forma digitale, sia il video sia l'audio.

Mouse e tastiera USB

Sicuramente, la prima cosa che cercherete di fare con il Raspberry è quella di renderlo il più simile possibile al vostro PC di casa. Per la connessione di monitor è tastiera non dovreste avere alcun problema; vi basterà collegare quelli di cui già disponete per il vostro PC desktop. Nel caso in cui decidiate comunque di acquistarne di nuovi potreste pensare di acquistare, con pochi euro, una mini-tastiera USB e un mi-

Figura 21: USB HUB, 2.0, powered (costo: 40-50 euro circa) [5].

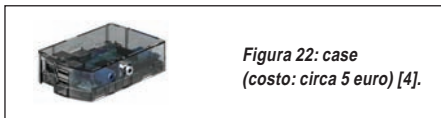



Figura 22: case (costo: circa 5 euro) [4].

ni-mouse USB **(figura 13)** in perfetto stile pocket Raspberry.
Nel caso in cui lavoriate con il Raspberry Pi Model A invece, disponendo di una sola porta USB dovrete munirvi di un HUB USB powered. In alternativa è anche possibile sostituire i due con una mini tastiera + touchpad in modo da necessitare di una sola porta USB del Raspberry e di lasciare la seconda porta USB (Modello B) libera per la connessione di un disco di memoria di massa esterno.

Adattatore PS2-USB

Con la diffusione dello standard USB sono orai divenuti poco diffusi i mouse e le tastiere PS2, tuttavia se qualcuno possiede un mouse o una tastiera PS2 può procurarsi un cavo adattatore 2xPS2-USB come quello mostrato in **figura 14**. L'acquisto costa forse quanto quello di mouse e tastiera ma consente di ottenere un ulteriore vantaggio, ovvero quello di utilizzare per mouse e tastiera una sola porta USB del Rasbperry Pi.

Cavo HDMI-DVI

Questo è il cavo sarà utile alla maggior parte di coloro che si cimenteranno con Rasbperry Pi in virtù del fatto che la maggior parte dei monitor per PC diffusi oggi presenta ingresso VGA e DVI. Un tale cavo adattatore consente infatti di adattare l'uscita video HDMI del Raspberry Pi all'ingresso video DVI del monitor.

Adattatore video HDMI-DVI

Un adattatore di questo tipo può essere connesso all'uscita HDMI del Raspberry Pi in modo da convertire quest'ultima in DVI femmina. Se disponete di un monitor con ingresso DVI, disponete sicuramente anche del relativo cavo DVI maschio-maschio. Un capo di questo va connesso al monitor e l'altro al connettore DVI femmina del suddetto adattatore.
In alternativa, è possibile ottenere lo stesso risultato connettendo un cavo HDMI maschio-maschio sull'uscita HDMI del Raspberry Pi; all'estremità libera del cavo connetterete un adattatore HDMI femmina-DVI maschio come quello riportato un **figura 17**. Quest'ultimo andrà quindi connesso direttamente all'ingresso DVI femmina del monitor.

Cavo video-composito

Il cavo video-composito consente di connettere direttamente l'uscita video-composita del Raspberry Pi all'ingresso video-composito di dispositivi TV muniti di questo tipo di ingresso video (quello caratterizzato dal connettore di forma tonda e colore giallo). Si tratta del classico cavo RCA-RCA **(figura 18)**

Adattatore da video-composito a SCART

Questo adattatore può risultare utile nel caso in cui vogliate utilizzare un monitor TV che non dispone né di ingresso video-composito, né di ingresso HDMI o DVI. In queste condizioni, tutti i precedenti adattatori e cavi video risultano inutilizzabili. Quasi certamente tuttavia, il dispositivo TV è in questo caso relativamente datato e dispone comunque di presa SCART. In queste condizioni, potrete utilizzare un cavo Video-Composito associato a un convertitore da video-composito a SCART per interfacciare il dispositivo TV con il Raspberry.

Cavo jack audio

L'uscita audio è predisposta per accogliere un jack stereo 3,5 mm come quello che trovate tipicamente su cuffie o casse amplificate. Per questo motivo non dovreste avere bisogno, per l'ascolto dell'audio di alcun cavo accessorio. Inoltre, va osservato che nel caso in cui si impeghi la connessione HDMI verso un monitor TV (attenzione, non HDMI+convertitore HDMI-DVI) non dovrebbe essere necessario utilizzare l'uscita audio analogica jack essendo questa già presente in digitale sulla connessione HDMI. L'HDMI è infatti un'interfaccia non compressa completamente digitale che trasporta sia il segnale audio sia il segnale video e che presenta il pregio di essere retro-compatibile con l'interfaccia digitale DVI che tuttavia è in grado di trasportare il solo segnale video.
Se si impiega, come uscita video, quella video-composita, è necessario impiegare come uscita audio quella jack 3,5 mm per la connessione allo stesso apparecchio

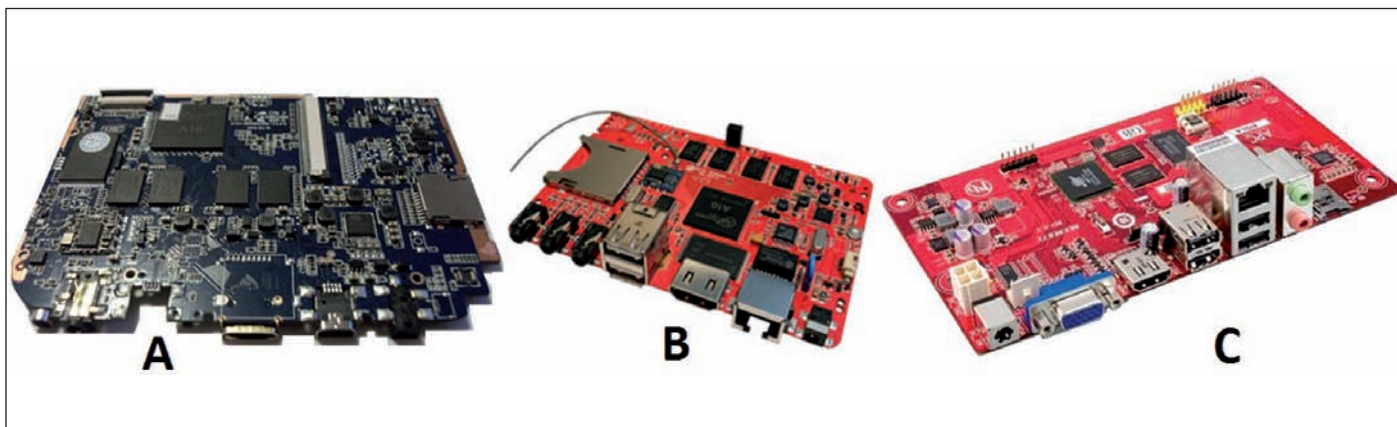



Figura 23: A) Gooseberry, un rivale di Raspberry Pi con Android [12]; B) Aackberry A10-Dev Board. 1 GB RAM, 4 GB internal flash, 2 porte USB 2.0, Ethernet 10/100Mbit, w-ifi: 802.11 b/g/n, video HDMI out, composite-video out, audio in minijack, audio out minijack [13], [14]; C) VIA PC-8750.

Figura 24: MK802 Android 4.0 Mini PC [20].

Figura 25: Cotton Candy (misura soli 8 cm e costa 199 euro).

TV, a una cuffia oppure a un altoparlante amplificato. La connessione audio allora, mentre è diretta negli ultimi due casi, necessita di un apposito cavo che consente il passaggio da jack 3,5 mm a doppio connettore RCA (con i classici colori standard rosso e bianco) nel primo caso.

SD Card

Il Raspberry Pi, come molti SBC (Single Board Computer) non è dotato di memoria di massa on board ed effettua pertanto il boot del sistema operativo direttamente da SD card **(figura 19)**. La scheda, con il sistema operativo precaricato, può essere preparata preventivamente secondo semplici procedure che saranno illustrate in successivi articoli di carattere maggiormente operativo. In alternativa, tra gli accessori per Raspberry Pi resi disponibili dai fornitori ufficiali Farnell e RS Components, è possibile acquistare la SD card con sistema operativo preinstallato. Per questo motivo, prima di procedere all'acquisto della scheda, ammesso che non ne abbiate già una disponibile, è necessario stabilire se si intende preparare da sé il caricamento sulla stessa del sistema operativo o acquistare una SD card già pronta. Il consiglio è quello di imparare a seguire la prima di queste due strade, sia perché così facendo imparerete ad avere un controllo maggiore dei vostri progetti sviluppati su Raspberry Pi, sia perché potrete riutilizzare la stessa SD card per provare diversi sistemi operativi in modo da confrontarne direttamente pregi e difetti. Una volta caricato il sistema operativo è possibile gestire un HD USB ma in ogni caso non è possibile eseguire il boot senza SD card. L'accensione del Raspberry Pi avviene, una volta inserita nel suo alloggiamento la scheda SD card con il sistema operativo precaricato, semplicemente alimentando la scheda. Analogamente lo spegnimento avverrà disalimentando la board.

Una scheda SD da 4 GB è certamente sufficiente per cominciare a lavorare con Raspberry, tuttavia disporre fin dall'inizio di una capacità superiore, per esempio 8 GB, non è sbagliato; la capienza dovrebbe in qualche modo risultare adeguata al proprio impiego ma nel dubbio è bene non limitare troppo le future possibilità. Sulla SD card va copiata l'immagine di una distribuzione Linux quale Debian Squeeze, Fedora, Arch Linux.
In **figura 19** abbiamo riportato modelli di SD Card rispettivamente di classe 6 e classe 4. Se non ne disponete e dovete acquistarne una senza SO precaricato, il consiglio è di acquistarla di classe non inferiore a 4 o 6: la sua velocità finirà prima o poi per influenzare inevitabilmente le prestazioni generali del vostro sistema Raspberry.
Nota: ricordiamo che le schede di memoria Flash SD sono classificate in base alla velocità che sono in grado di garantire e più precisamente in base al cosiddetto Minimum Speed Ratings: Class 2 = 2 MB/s; Class 4 = 4 MB/s; Class 6 = 6 MB/s; Class 10 = 10 MB/s.

Card reader

Nel caso in cui decidiate di caricare voi il SO sulla SD card necessiterete sicuramente di un card reader. Nel caso in cui il vostro PC non disponga di un card reader integrato dovrete acquistarne uno esterno USB **(figura 20)**.

Connettività

La connettività a una rete LAN o a Internet è possibile via cavo attraverso un cavo Ethernet/LAN (classico cavo di rete con connettore RJ45) o, in alternativa, attraverso una rete Wi-Fi, mediante un adeguato adattatore USB Wi-Fi. Al fine di sgombrare da subito il campo da dubbi, è opportuno segnalare che la porta Ethernet del Raspberry Pi è auto-sensing per cui può essere connessa a un computer oppure a uno switch o a un router usando sempre il classico cavo ethernet diritto senza bisogno di procurarsi alcun cavo incrociato.

HUB USB

Il numero di porte USB disponibili può facilmente essere aumentato mediante l'impiego di un HUB USB. L'unica accortezza opportuna in questa operazione è quella di impiegare un Hub USB di tipo powered, cioè alimentato. Questo darà la certezza che il dispositivo connesso possa ricevere la sufficiente potenza per il suo corretto funzionamento senza gravare sul piccolo Raspberry.
Lo standard USB raccomandato è il 2.0 **(figura 21)** poiché la versione 1.1, sebbene adeguata alla connessone di mouse e tastiera, potrebbe rivelarsi poco veloce per lo scambio dati con altri dispositivi e divenire pertanto un collo di bottiglia nella velocità dell'intero sistema.

Case

Il Raspberry Pi, nella sua configurazione standard attuale, viene fornito come scheda nuda, cioè senza alcun case di protezione. In linea di massima, questo non pone particolari limiti al suo impiego, tuttavia è sempre opportuno assicurarsi che esso non venga a contatto con liquidi, solidi conduttori ed elettricità statica.
Chi è abituato a maneggiare schede elettroniche sa bene come evitare danni, tuttavia trattandosi di una board che ci si aspetta che resti sulla scrivania o sul banco di lavoro dello sviluppatore per diverso tempo è opportuno che sia dotata di un case di protezione anche minimale purché efficace. Il costo contenuto della board non deve infatti significare libertà di "maltrattarla". È possibile realizzare case artigianali oppure acquistare uno dei case già realizzati ad hoc e messi a disposizione dagli stessi distributori che distribuiscono la board **(figura 22)**.
In ogni caso, fino a che non disporrete di un case adeguato è opportuno, almeno, tenere la scheda sopra la bustina antistatica di protezione che la contiene all'acquisto.

ELENCO COMPONENTI, ADATTATORI E CAVI ACCESSORI

E opportuno riassumere quanto esposto, in merito agli accessori, adattatori e cavi necessari per l'allestimento di una

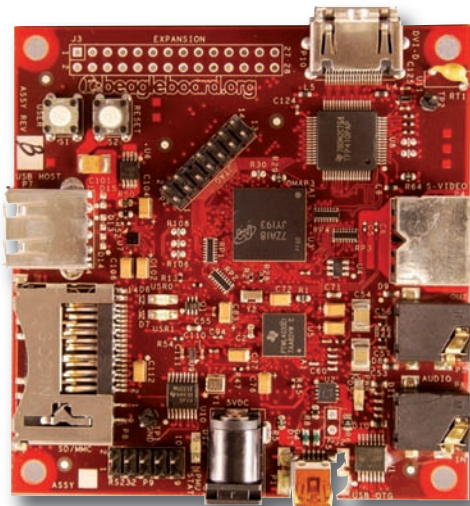
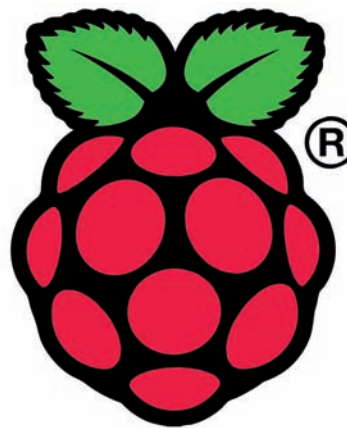
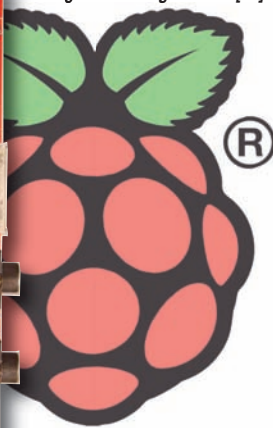



Figura 26: Beagle Board [11].

postazione di lavoro per i nostri esperimenti con Raspberry, in una tabella di riferimento **(tabella 5)**. Al fine di rendere la stessa quanto più possibile utile da un punto di vista pratico al lettore, si riporta indicazione per ciascuna parte rispettivamente di: figura di riferimento, codice prodotto di un distributore ufficiale di Raspberry Pi (in questo caso abbiamo fatto riferimento al distributore Farnell **[5]**) e relativo costo di acquisto al momento della stesura del presente articolo.

CHE COSA È POSSIBILE FARE CON RASPBERRY PI

Fino a questo momento abbiamo esplorato il Raspberry Pi e ne abbiamo indagato le caratteristiche, abbiamo anche intuito che cosa sia possibile fare con Raspberry Pi e abbiamo compreso che è possibile usarlo anche come un comune PC di casa per scrivere documenti (word-processing), costruire fogli di calcolo, disegnare, navigare in Internet; soprattutto abbiamo però scoperto, per la gioia di appassionati di elettronica e soprattutto dell'informatica open source, che può essere utilizzato come piattaforma embedded di programmazione e/o di sviluppo. Volendo esplicitare i possibili impieghi possiamo prendere quindi a riferimento il seguente elenco:

- piattaforma di programmazione/sviluppo embedded;
- web browsing;
- PC Media Center (eventualmente utilizzando una porta USB per il collegamento di una chiavetta WiFi);
- NAS;
- video games (grazie alla gestione video in alta definizione);
- server di rete;
- centralina di controllo;
- impiego come PC tradizionale.

Queste sono solo alcuni dei possibili usi, ma se ne potrebbero pensare molti altri più o meno utili, più o meno complessi e più o meno affascinanti; per esempio non è difficile pensare che qualcuno possa cercare di combinare il Raspberry con un display touch-screen con il fine di realizzare, artigianalmente, un tablet tanto funzionale quanto economico.

PROGETTI CONCORRENTI A RASPBERRY PI

L'impiego dei computer si è evoluto nel tempo da concetto di PC da ufficio a quello di computer per il controllo di processi e applicazioni industriali. L'evoluzione dei PC embedded ha portato a inglobare caratteristiche e funzionalità sempre più evolute a tutto favore del time-to-market. Il tempo e lo sforzo per lo sviluppo di applicazioni embedded si sono in questo modo via via ridotti. Questo processo ha comportato, specie negli ultimi anni, un proliferare di PC embedded dalle forme, dimensioni, prestazioni e prezzi disparati. In molti casi si tratta di piattaforme basate su Linux, che rappresenta un'alternativa valida e robusta a Microsoft Windows, oltre che open source, e adattata alle applicazioni embedded. Concetti di open hardware e open software si sono sempre più fatti strada ed hanno indirizzato lo sviluppo dei sistemi embedded verso una progressiva riduzione di costi, riduzione dei tempi di sviluppo e raggiungimento di rapporti qualità-prezzo e affidabilità sempre migliori. L'obiettivo che sta all'origine dello sviluppo del progetto Raspberry Pi non è in realtà quello di rivolgere il prodotto alle applicazioni di controllo industriale, tuttavia ne condivide, dal punto di vista tecnico, molti aspetti.

L'uscita sul mercato di Raspberry Pi ha dato, inoltre, ulteriore impulso alla diffusione di PC embedded dalle potenzialità più o meno ampie. Molte alternative non sono affatto economiche pur essendo maggiormente performanti; altre, pur caratterizzate da costi più contenuti, anche se difficilmente paragonabili a quelli davvero ridotti di Raspberry, differiscono in maniera più o meno marcata per caratteristiche, equipaggiamento e prestazioni.

Raggiungere il rapporto tra flessibilità e prestazioni da una parte e prezzo dall'altra offerto dal Raspberry non è infatti impresa facile. Basti pensare che, a un costo circa uguale a quello di Arduino2009, con Raspberry abbiamo a disposizione un computer con processore ARM11 a 32 bit cui un microcontrollore non è paragonabile. Completamente diverse sono invece le valutazioni se si confronta Raspberry Pi con single Board Computer come Beagle Board, basata su ARM Cortex A8@1GHz **[11]** ma va detto che completamente diversi sono, in questo caso, anche i costi (superiori a 100 euro).

Raspberry Pi dimostra, in ogni caso, che per avere un PC embedded utile per una grande varietà di cose e anche come piattaforma di sviluppo in ambiente open source, in sintesi qualcosa che sia anche adeguata alle esigenze di noi appassionati di elettronica e programmazione, non occorre in realtà spendere una cifra elevata.

Inoltre, anche se non si tratta di un PC completo in grado di utilizzare in maniera fluida qualunque videogioco, è più che sufficiente per svariate applicazioni, compresa la navigazione su Internet.

Volendo fare un parallelismo, si potrebbe azzardare l'ipotesi che Raspberry Pi diventi in breve tempo quello che per le generazioni di ragazzi degli anni Ottanta sono stati i primi Amiga, PC da studio utili anche all'apprendimento dell'informatica e della programmazione.

In conclusione, il Raspberry Pi non può

TABELLA 5

ACCESSORIO/CAVO/ADATTATORE	FIGURA DI RIFERIMENTO	CODICE FARNELL [5]	COSTO (EURO)
Alimentatore micro USB			
(5 V, I >= 700 mA)	Figura 9	2081769	8,2
Alimentatore con uscita su presa USB	Figura 10	2100375	8,2
cavo USB A male-micro B male (costo 3 euro) [5]	Figura 11	2085464	1,6
Cavo HDMI maschio-maschio			
0,5 m	Figura 12	2113608	1,2
1,0 m		2113609	2,5
2,0 m		2113607	1,9
5,0 m		2113610	4,1
10,0 m		2113611	8,7
Mini Mouse USB	Figura 13	2113619	5
Mini Tastiera USB	Figura 13	2113614	10
Cavo Adattatore 2xPS2-USB	Figura 14	4407908	30
Adattatore HDMI Femmina-DVI Maschio	Figura 17	1428271	8,2
HDMI Maschio-DVI Femmina	Figura 16	1428272	8,2
Cavo Adattatore da HDMI a DVI-D (L = 2 m)	Figura 15	1813133	10
Cavo Video-Composito			
Cavo USB A-Micro USB B;			
1 m	Figura 18	2085464	1,6
1,8 m		2115733	3,2
5,0 m			
Cavo Ethernet diritto (cat 5E)		1734943	1,2
1 m		1734946	2,8
3 m	-	1734948	3,7
		1734948	7
Adattatore da Video-composito a SCART	-	-	-
Cavo jack Audio	-		
SD Card con 4GB cl. 4 con Debian 6 preinstallato	Figura 19	2113756	10
Card Reader	Figura 20	2081767	3,8
HUB USB powered	Figura 21	3934240	49
		8704341	27
		2115058	10
Case trasparente [4]	Figura 22	-	5

Tabella 5: elenco degli accessori che potrebbero risultare utili per l'allestimento della propria postazione Raspberry Pi, codici prodotti e relativi costi di acquisto al momento della stesura dell'articolo (distributore Farnell-[5]).

certo essere paragonato a una CPU potente come le moderne CPU a 64-bit, ma nella sua fascia di mercato e per le applicazioni per le quali è stato pensato si fa rispettare; inoltre costa davvero poco ed è piccolo come poche schede concorrenti.
Tra i possibili rivali di Raspberry si sta ponendo Gooseberry **(figura 23)** che, a differenza delle soluzioni che montano distribuzioni Linux, si basa sul sistema operativo Android Ice Cream Sandwitch. Le caratteristiche della scheda sembrano indicarla come una scheda più performante di Raspberry ma a un costo maggiore (una cinquantina euro circa). Non è inoltre da trascurare la scarsa flessibilità attuale dovuta al fatto che attualmente Android ICS è l'unico sistema operativo supportato. Le dimensioni sono inoltre superiori a quelle mostrate da Raspberry.
MK802 **(figura 24)** è invece una SBC progettato intorno a un processore ARM Cortex A8 con frequenza di clock di 1,5 Ghz, equipaggiato con 512 MB di RAM e 4 GB di memoria interna espandibile via microSD. Dotato di interfaccia Wi-Fi, è equipaggiato con una porta HDMI e fun-

ziona con Android 4.0, il tutto a un costo di circa 55-60 euro.
Il successo del RaspBerry Pi ha insomma innescato l'uscita sul mercato di alternative simili con costi al ribasso. VIA PC-8750, per esempio **(figura 23)** è un altro potenziale concorrente di Raspberry Pi basato però su Android, equipaggiato con processore ARM11 da 800MHz, 2 GB memoria flash e 512 MB di RAM DDR3, quattro porte USB 2.0 e uno slot microSD. Simile come concezione a Raspberry Pi, può essere collegato a un monitor oppure a un apparecchio TV tramite VGA o HDMI ed è dotato di connettività Ethernet e prese jack da 3,5 mm per l'ingresso e l'uscita audio. Supporta OpenGL e riproduce filmati a 720p.

In pratica si tratta di una piattaforma simile a quelle che troviamo su uno smartphone o un tablet.
L'OS è un Android 2.3 modificato per accogliere mouse e tastiera. Il costo è di circa 40 euro. Non solo il costo è superiore a quello di Raspberry, anche le dimensioni (17x8,5 cm) risultano più ingombranti. Inoltre, APC è poco adeguato a chi vuole programmare o sviluppare progetti embedded, a meno di non riuscire a farci girare qualche distribuzione Linux.
Alto mini-PC interessante è Candy Cotton **(figura 25)**. Costruito intorno a un processore ARM Cortex A9 con frequenza di clock pari ad 1,2 GHz, si basa su Android 4.0 ed Ubuntu, garantisce 1 GB di memoria DRAM DDR3 e altri 64 possono essere equipaggiati mediante scheda microSD. Integra processore grafico Mali 400 MP, connettività alla rete tramite interfaccia Bluetooth e Wi-Fi, USB 2.0 e porta HDMI. Il suo prezzo è tuttavia sostenuto e si aggira intorno ai 150 euro.
Una menzione a parte merita probabilmente BeagleBoard **(figura 26)**, un SBC in grado di offrire prestazioni e connettività paragonabili a quelle di un laptop. È progettato intorno a un SoC di Texas Instruments, l'OMAP3530, che integra un processore superscalare ARM Cortex TM-A8 da 600 MHz, coprocessore, DSP TMS320C64x+TM da 430 MHz e acceleratore grafico 2D/3D OpenGL ES 2.0. Ovviamente, se confrontata al Raspberry Pi la differenza di costo appare evidente (costa intorno ai 100 euro).
Confrontato con le alternative in commercio ci sembra certamente tra le soluzioni più flessibili e performanti. Di seguito le sue principali caratteristiche:

- OMAP3530;
- chip DVI (TFP410);
- DVI-D;
- JTAG 14 pin;
- connettore di espansione (I²C, I2S, SPI, MMC/SD), non montato nella Rev B5.;
- tasto user (permette di variare l'ordine dei dispositivi con cui effettuare il boot);
- tasto reset;
- USB 2.0;
- SD/MMC+;
- RS-232;
- ingresso audio (stereo);
- uscita audio (stereo);
- S-Video.

CONCLUSIONI

In questa seconda puntata dedicata a Raspberry Pi ne abbiamo analizzato l'hardware "passandolo sotto la lente di ingrandimento". Abbiamo anche indagato sugli accessori (adattatori, cavi, periferiche esterne) che non essendo previste come componenti in dotazione è necessario acquistare separatamente (non necessariamente tutti) se si vuole allestire una postazione Raspberry Pi completa e funzionale per i nostri esperimenti e per lo sviluppo dei nostri progetti. Siamo infine tornati ad inquadrare Raspberry Pi nell'ambito delle schede di sviluppo per sistemi embedded SBC (Single Board Computer) proponendo un confronto sia tecnico sia di costo con altre schede e progetti concorrenti. Anche acquistando tutti gli accessori del caso, la lista complessiva della spesa può essere contenuta abbastanza agevolmente al di sotto dei 100 euro.
Nelle prossime puntate focalizzeremo maggiormente la nostra attenzione su aspetti operativi ovvero sull'impiego pratico di Raspberry Pi. Ci occuperemo pertanto sia dell'installazione dei sistemi operativi sulla SD card di boot sia dello sviluppo di progetti più o meno complessi basati sulla programmazione di Raspberry Pi. ❑

CODICE MIP 2828844

STAMPA 3D con KISSlicer

Uno dei passi da effettuare nella realizzazione di un oggetto mediante una stampante 3D è trasformare il file 3D in percorsi GCODE che la stampante seguirà per realizzare l'oggetto. Per fare questa operazione esistono diversi programmi software, in questo articolo vedremo come KISSlicer esegue ottimamente questo compito.

Riprendiamo con questo numero l'argomento "stampa 3D" aperto con il numero di luglio/agosto di Fare Elettronica. Come visto nell'articolo introduttivo, la realizzazione di un oggetto tridimensionale passa attraverso varie fasi, tra cui la conversione del modello 3D in un file STL e la conversione di quest'ultimo in percorsi GCODE.
Un file GCODE è un file di testo in cui ogni riga contiene informazioni sul percorso che la macchina dovrà seguire, oltre ad altri comandi quali la velocità e la quantità di materiale da estrudere. In questo articolo approfondiremo proprio questo aspetto: la conversione di un file STL in GCODE, un'operazione molto delicata da cui dipende la qualità e la concretezza dell'oggetto stampato. La conversione viene fatta attraverso software specifici, per la nostra trattazione utilizzeremo KISSlicer, viste le sue grandi potenzialità e la facilità d'uso. Per rendere "pratica" la trattazione considereremo un oggetto (ad esempio la molletta di **figura 1**) e vedremo come i vari parametri influenzano il risultato della stampa 3D.

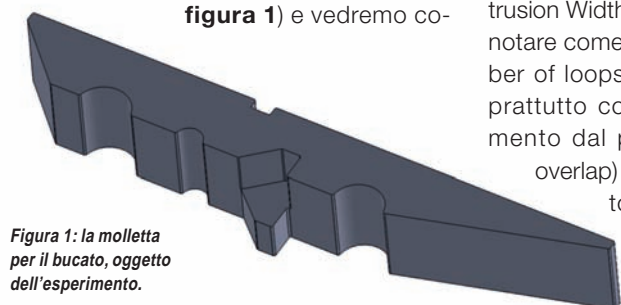



Figura 1: la molletta per il bucato, oggetto dell'esperimento.

Risultato	Infill Extrusion Width	Infill %	Number of loops	Loop/Solid infill overlap
	0.2	50	2	80
	0.4	50	3	1
	0.5	50	3	1
	1	50	3	1
	0.2	20	2	0.8

Figura 2: il risultato della stampa in funzione dei parametri impostati.

LA PERCENTUALE DI RIEMPIMENTO

Vediamo ora un primo esempio che illustra l'influenza della percentuale di riempimento (Infill %) e dello spessore (Infill Extrusion Width) dello stesso. Si potrà anche notare come il numero di perimetri (Number of loops) influenzi la struttura e soprattutto come la distanza del riempimento dal perimetro (Loop/Solid infill overlap) possa rendere l'oggetto molto più solido.
Nella **figura 2** è riportato, per ogni riga, il risultato della stampa in funzione dei parametri suddetti. Nella prima riga si vede chiaramente che il perimetro è troppo sottile, il riempimento è staccato dal perimetro ed è anch'esso molto sottile. Questo genererà un oggetto leggero ma non strutturalmente resistente con tempo di realizzazione contenuto e consumo di materiale molto ridotto. Nella quarta riga si vede chiaramente che il perimetro è corretto, il riempimento si sovrappone al perimetro che in questo caso è molto compatto. Questo genererà un oggetto pesante e strutturalmente molto resistente con tempo di realizzazione molto lungo e consumo di materiale elevato. Nella seconda e terza riga si vede chiaramente che il perimetro è corretto, il ri-

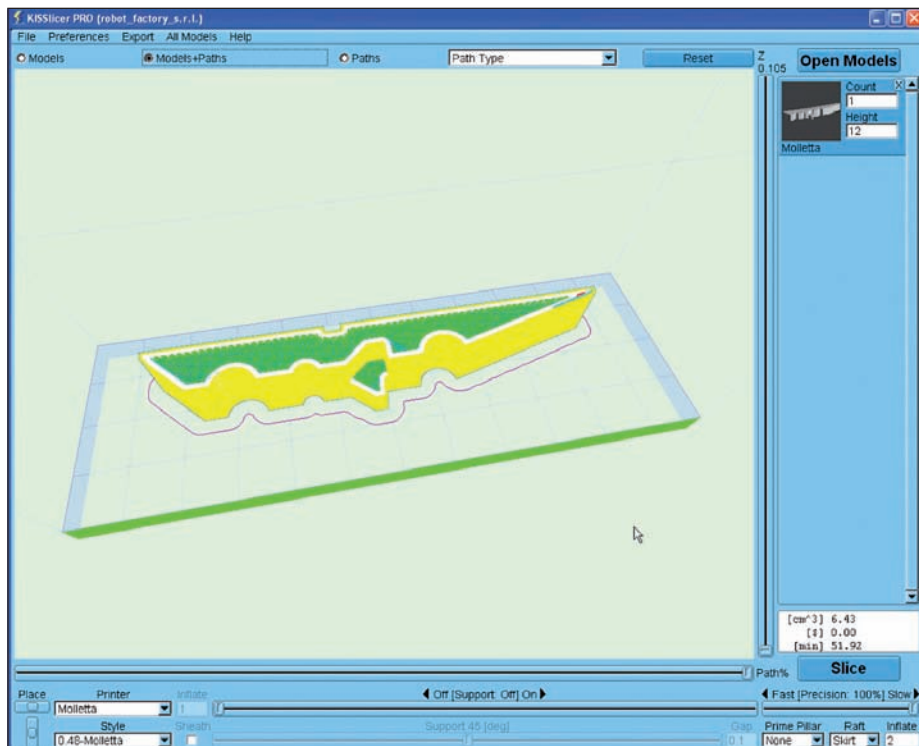



Figura 3: schermata principale di KISSlice.

empimento si sovrappone al perimetro ed è sufficientemente compatto e solido. Questo genererà un oggetto relativamente leggero e strutturalmente resistente con tempo di realizzazione medio e consumo di materiale contenuto. Nella quinta riga si vede chiaramente che il perimetro è troppo sottile, il riempimento è staccato dal perimetro e non è omogeneo. L'oggetto sarà estremamente leggero ma strutturalmente molto fragile con tempo di realizzazione molto contenuto e consumo di materiale molto ridotto.

INTERFACCIA

In **figura 3** è riportata la schermata principale del programma. Si fanno notare i seguenti campi:

- Printer;
- Style;
- Raft.

Nella configurazione si sono creati i profili Molletta (Printer) e 0.48-Molletta (Style) che consentiranno di utilizzare le impostazioni anche per altri oggetti, oppure per poter rielaborare la nostra molletta in un momento successivo anche a distanza di molto tempo. Si fa notare inoltre che non è stata utilizzata la funzione di supporto (Support). Nelle figure 4, 5, 6, 7, 8 e 9 sono riportati i parametri impostati rispettivamente per i campi Style, Material, Printer, Speed, Extruders, G-Code. Nel campo Material si notino i parametri Destring [mm] Suk e Prime: questi due parametri consentono di eliminare i noiosi fili che si possono formare per effetto del gocciolamento dell'ugello nel passaggio da un punto all'altro. Nel campo Printer è possibile specificare le dimensioni del piano di lavoro, mentre nel campo Speed si possono impostare le varie velocità. Si ricorda che le velocità sono espresse in Kisslicer come mm/sec, mentre in Mach3 (g-code) queste
verranno espresse in mm/min. Per una rapida conversione:

- mm/sec x 60 = mm/min;
- mm/min / 60 = mm/sec.

Per esempio:
X,Y Travel Speed

- 50 mm/sec x 60 = 3000 mm/min.

Nel campo Extruders si seleziona l'asse A, nel campo G-code il parametro Firmware Type si è impostato su 5D – absolute E in modo che l'estrusore venga gestito con il motore passo passo dell'asse A e con coordinate assolute. Per ottenere un oggetto particolarmente definito, si dovrà scegliere la precisione maggiore (100%) e la velocità più bassa tra quelle impostate nella configurazione Speed **(figura 10)**.

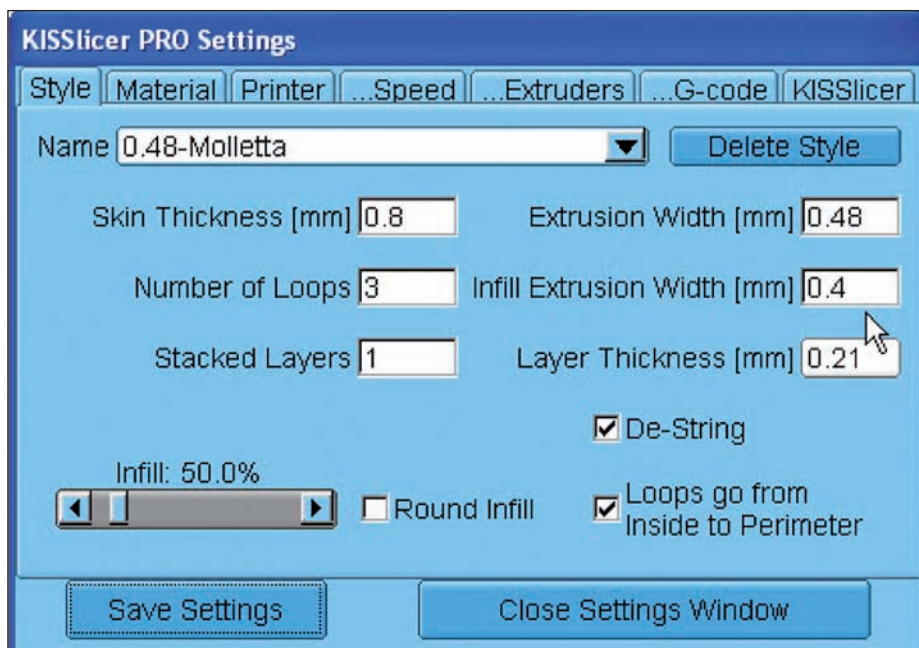



Figura 4: parametri impostati per il campo Style.

KISSlicer PRO Settings

Style | Material | Printer | ...Speed | ...Extruders | ...G-code | KISSlicer

Name ABS rosso — Delete Material

NOTE: Select a given material on the [Extruders] tab.

Calculate — Fan Always OFF — Color

0 — Warm-Up Factor — Extruder RPM Min 1 . . . . Max 75

Fiber Dia [mm] 3 — Flow Tweak 1

Calculate — Min Layer Time [s] 10 — Wipe [mm] 10

0 — $ / cm^3 — Destring [mm] Suck 0.5 . . . . . Prime 0.5

Destring Speed [mm/s] 15 — Destring Min 0

Temp [C] 10 — Keep-Warm 10 — First Layer 10 — Bed 3

Save Settings — Close Settings Window

Figura 5: parametri impostati per il campo Material.

K-EXTRACTOR

K-Extractor è un piccolo programma di utilità scritto in Perl da Max72 (Massimo) e pubblicato nel forum di Kisslicer [1]. Per poter utilizzare il programma originale è necessario avere installato sul computer il linguaggio Perl [2].
Poiché non tutti hanno dimestichezza con la riga di comando (prompt DOS) e tanto meno con il linguaggio Perl, Robot Factory, per permetterne l'uso ai propri clienti, utilizzando il compilatore Perl2exe [3] ha trasformato il file originale in eseguibile, apportando inoltre alcune modifiche. Il programma K-Extractor permette di estrarre dal file G-code generato da Kisslicer, i file di configurazione (file *.ini). Questo consente di rielaborare, anche a

Figura 6: parametri impostati per il campo Printer.

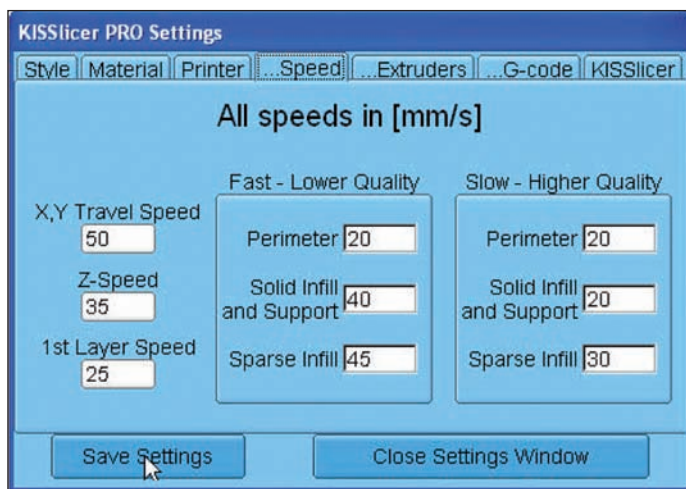



Figura 7: parametri impostati per il campo Speed.

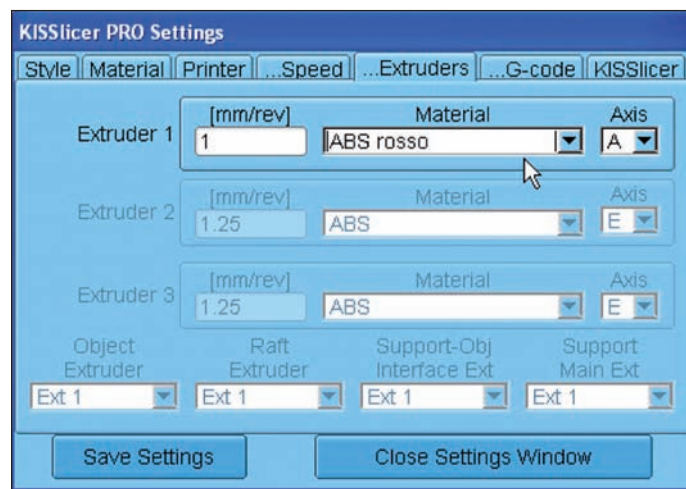



Figura 8: parametri impostati per il campo Extruders.

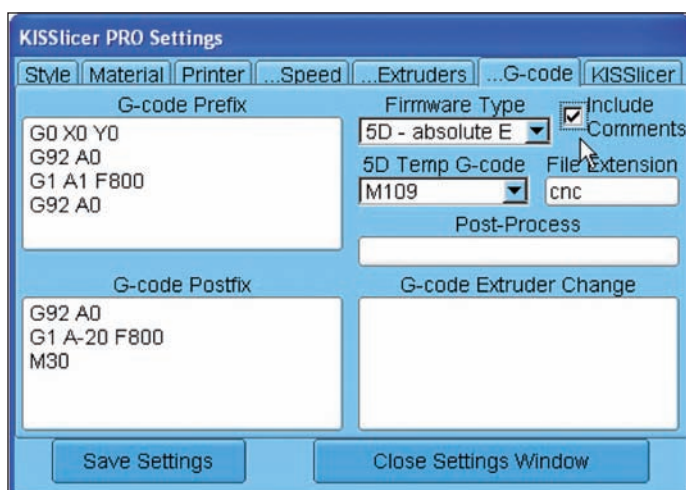



Figura 9: parametri impostati per il campo G-Code.

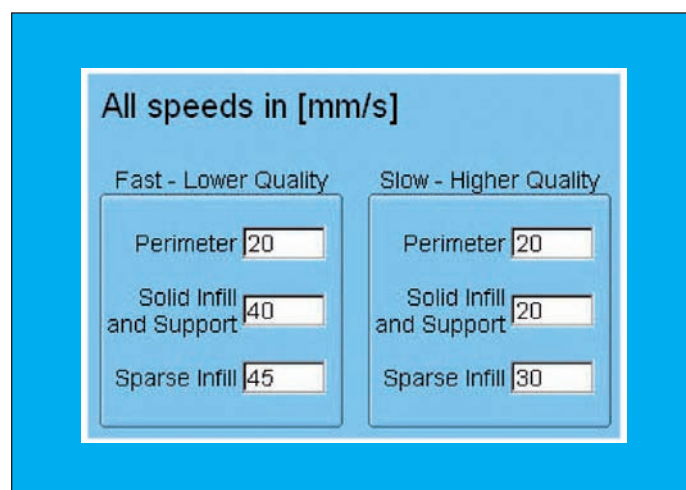



Figura 10: impostazioni di velocità e precisione.

distanza di tempo, il file STL partendo dalla configurazione impostata. Ci si ritrova cioè, nella condizione e con le configurazioni di quando si era generato il codice. K-Extractor si può scaricare dal sito all'indirizzo indicato dal riferimento **[4]**.

Vediamo allora un esempio pratico, utilizzando ancora la molletta e supponendo che il programma sia stato scompattato nella directory C:\K-Extractor. Per chi ha un po' di dimestichezza con il computer e con il relativo prompt del Dos, dalla taskbar con il bottone Start selezionare Run quindi digitare cmd per aprire una finestra DOS. Spostarsi nella cartella K-Extractor, eseguire il programma inserendo come parametro il nome del file da elaborare (nel nostro esempio Extract Molletta.cnc) e premere Invio. Nella cartella C:\K-Extractor si saranno creati i relativi file di configurazione per Kisslicer. Salvare in una directory di appoggio i file (*.ini) presenti nella directory C:\Kiss per non sovrascriverli. Copiare i file creati con K-Extractor nella cartella C:\Kiss.

Con Kisslicer, aprire il file Molletta.stl presente nella cartella C:\K-Extractor\Molletta. Nella cartella C:\K-Extractor\Molletta\Impostazioni, si trovano le immagini per un eventuale confronto dei parametri di configurazione ottenuti con il programma.

Nei prossimi numeri di Fare Elettronica scenderemo nel dettaglio della stampa degli oggetti. Seguiteci! ❑

CODICE MIP 2828848

PER *approfondire...*

- [1] http://kisslicer.com/forum/index.php?topic=171.0
- [2] http://www.activestate.com/activeperl/downloads
- [3] http://www.indigostar.com/perl2exe.php
- [4] http://www.robotfactory.it/Robotfactory/K-Extractor.zip

Corso di Mikropascal
(parte seconda)

L'AMBIENTE di sviluppo

In questa seconda puntata del corso cominceremo a utilizzare l'ambiente di sviluppo per la creazione dei nostri primi progetti. In particolare, vedremo come programmare il PIC in maniera da pilotarne le porte di uscita. Nel far ciò prenderemo confidenza, in particolare, con l'editor di MikroPascal e con le funzioni di compilazione e linking integrate. Per una più facile comprensione, non mancheremo di dare un primo sguardo a problematiche che riguardano la connessione fisica del controllore al mondo esterno.

Nella prima puntata del corso MikroPascal per PIC abbiamo presentato al lettore l'ambiente di sviluppo inquadrando il tutto all'interno del più ampio panorama dei controllori PIC di Microchip. Attraverso il download della versione demo del pacchetto MikroPascal (ver. 5.61) **[3]** e la sua installazione ci siamo dotati dello strumento principale per affrontare il nostro viaggio nella programmazione Pascal dei controllori PIC.
In questa seconda puntata approfondiremo la conoscenza pratica dell'IDE MikroPascal utilizzandolo per la creazione dei nostri primi progetti che avranno le seguenti finalità:
1. imparare a impostare e a creare un nuovo progetto;
2. prendere confidenza con alcune funzioni dell'editor che facilitano la stesura del codice;
3. prendere confidenza con la struttura generale di un codice Pascal;
4. imparare l'utilizzo delle prime parole riservate;
5. imparare la prima è più ricorrente struttura di controllo di un codice MikroPascal per PIC;
6. imparare a configurare le porte del PIC come uscite utilizzando istruzioni adeguate;
7. imparare a pilotare le uscite del controllore per controllare circuiti esterni impiegando istruzioni MikroPascal;
8. cominciare a utilizzare, con una certa disinvoltura, dati espressi in notazione rispettivamente decimale, esadecimale e binaria;
9. imparare a compilare il codice di programma e a riconoscere gli errori più banali che possono far fallire la compilazione grazie all'analisi dei messaggi che il compilatore fornisce allo sviluppatore alla fine del processo di compilazione e linking.

In particolare, quanto richiamato ai punti 1, 2 e 9, aiuta lo sviluppatore, da un alto, a scrivere il codice di programma con relativa facilità e, dall'altro, a tenerlo costantemente sotto controllo durante il suo sviluppo.
Dovendo imbatterci in concetti quali memoria flash e I/O (Input/Output), daremo alcuni cenni alle risorse fisiche del controllore PIC e alla connessione circuitale del controllore stesso al mondo che lo circonda. Rimanderemo invece alle prossime puntate aspetti quali la programmazione fisica della memoria del controllore cioè il trasferimento del firmware (codice eseguibile) sulla memoria flash dello stesso, cenni all'architettura dei controllori PIC e alla circuiteria interna delle porte di I/O e discussioni o approfondimenti ulteriori su utility integrate nell'IDE.

PRIMA DI INIZIARE CON LA PRATICA

Durante il nostro percorso utilizzeremo controllori PIC di semplice reperibilità, prevalentemente il PIC16F877. La scelta è legata al fatto che si tratta di un controllore che presenta, come vedremo, una dotazione hardware più che sufficiente a condurre la maggior parte degli esperimenti che intendiamo mostrare in un corso di base come quello che stiamo intraprendendo. Dal sito del costruttore, cioè Microchip (www.microchip.com), è possibile effettuare il download del manuale di riferimento **[1]**. Si consiglia di tenerlo a portata di mano, per lo meno in

di ANTONIO GIANNICO

formato elettronico (non è il caso di procedere a una stampa cartacea visto che è composto da oltre duecento pagine).
Per il test pratico dei nostri esperimenti utilizzeremo, nella maggior parte dei casi, la scheda di sviluppo EasyPIC 3, ma ovviamente potremmo utilizzare agevolmente anche una qualunque delle versioni successive della stessa scheda, oppure potremmo montare direttamente i circuiti che mostreremo su bread-board e provvedere alla programmazione del PIC attraverso appositi programmatori. Su questo specifico argomento daremo alcune indicazioni ulteriori nel corso delle prossime puntate.
Il controllore PIC 16F877 lavora a tensione pari a 5 Vcc. L'aspetto alimentazione dei circuiti con cui avremo a che fare non dovrebbe costituire un problema. Se disponente di una scheda di sviluppo come EasyPIC disponente probabilmente anche di un alimentatore per alimentarlo. Si tratta di un alimentatore ovviamente a tensione DC superiore a 5 V, dal momento che la scheda EasyPIC è dotata di apposito regolatore on board. Se, in ogni caso, la vostra EasyPIC non dispone di alimentatore, ricordatevi che essa può anche essere alimentata direttamente dalla USB del vostro PC cui avrete modo di collegarla per la programmazione. Nel caso in cui non disponiate di EasyPIC ma di altre demo board per PIC valgono considerazioni simili. Nel caso invece in cui vogliate realizzare i circuiti che proporremo su una bread-board potrete reperire presso un qualunque negozio di elettronica un alimentatore a 5 V. La maggior parte dei circuiti che proporremo durante il corso infatti, non sono particolarmente complessi e una bread-board (e un programmatore) potranno, nella maggior parte dei casi, risultare più che sufficiente per i nostri scopi. Anche molti comuni alimentatori per telefonia cellulare sono in grado di erogare tensione di alimentazione pari a 5 V. Nel caso in cui disponiate invece di un vecchio alimentatore che fornisce tensioni maggiori, per esempio 8-9 V, potete pensare di utilizzare semplici regolatori di tensione come l'LM7805 o l'LM317 per ottenere i 5 V per l'alimentazione del controllore. Per chi dispone di un piccolo laboratorio hobbistico, e quindi di un alimentatore da banco con tensione di uscita DC regolabile, invece il problema non si pone.
Per completezza, ci sembra opportuno ricordare che esistono in letteratura diversi testi che trattano l'hardware e la programmazione dei controllori PIC di Microchip e che risultano particolarmente adatti per un approccio di base. Tra essi segnaliamo "Conoscere ed usare-PIC micro" degli autori T. Galizia e M. Del Corso, "Programmazione Basic per PIC" dell'autore G. Di Maria e "Pillole di microcontrollori PIC" dell'autore P. Rognoni, tutti editi da Inware Edizioni. Trovate maggiori informazioni in merito su www.farelettronica.com/libri.

TABELLA 1

PIC16F877 - CARATTERISTICHE FISICHE	
Frequenza operativa (clock)	**Fino a 20 MHz**
Reset (e delay)	**POR, BOR (PWR, OST) (*)**
Flash Program Memory (parole di 14 bit)	**8k**
Data Memory (byte)	**368**
EEPROM data Memory (byte)	**256**
Interrupt	**14**
Porte di I/O	**Porte A, B, C, D, E**
Timer	**3**
Moduli Capture/Compare/PWM	**2**
Comunicazione seriale	**MSSP, USART**
Comunicazione parallela	**PSP**
Conversione A/D a 10 bit	**8 ingressi**
Set di istruzioni	**35 istruzioni**

(*) Durante il corso avremo modo di scoprire il significato di questi acronimi

Tabella 1: principali caratteristiche fisiche del PIC16F877 [1].

PIC 16F877: CENNI ALLE RISORSE FISICHE INTEGRATE

Nel corso della prima lezione abbiamo fornito alcuni cenni sulla struttura interna dei controllori e dei PIC in particolare e abbiamo visto come un PIC integri in generale:
- CPU (Central Processing Unit) che esegue in successione le istruzioni di codice eseguibile;
- FLASH Program Memory per contenere il firmware (codice eseguibile);
- EEPROM (Electrical Erasable Programmable Read Only Memory) in grado di mantenere immagazzinati dati anche in assenza di alimentazione;
- RAM (Random Access Memory) contenente dati transitori;
- PORTE di I/O analogiche e digitali che rappresentano l'interfaccia con il mondo esterno;
- periferiche di comunicazione.

Si tratta di elementi distintivi tipici dei controllori e comuni praticamente a tutti i modelli di PIC. Differenze tra i diversi modelli sussistono soprattutto in termini di complessità della CPU, dimensione fisica di EEPROM, PROM, RAM, numero di pin e porte di I/O, tipo e numero di periferiche di comunicazione integrate (UART, USB, Ethernet, I2C, SPI, CAN ecc.).
Per entrare nel concreto, analizziamo brevemente questi aspetti per il PIC 16F877 riportando in **tabella 1** un riassunto delle sue principali risorse fisiche **[1]**.
Nel caso in cui vi troviate per le mani un differente modello di PIC potrete eseguirne un'analisi analoga, manuale alla mano, ed eventualmente un confronto con il PIC 16F877.

PACKAGE, PIN-OUT E PROGRAMMAZIONE FISICA DELLA MEMORIA FLASH

Prima di illustrare la nostra prima applicazione con MikroPascal, diamo un'occhiata ai pin del PIC16F877 e, in particolare, ai pin di I/O. In **figura 1** è riportato il pin-out del controllore. La versione di PIC16F877 che utilizzeremo per quanto riguarda il package è quella PDIP a 40 pin. I tre modelli di PIC presentati in **figura 1** rappresentano tutti lo stesso controllore e differiscono unicamente per il package. Il fatto di fare riferimento al package PDIP è legato alla sua caratteristica di essere più facile da utilizzare su una bread-board dal momento che le altre due versioni (QFP e PLCC) necessitano di montaggio superficiale. Il QFP ed il PLCC presentano 44 pin invece di 40 ma solo perché 4 di questi sono NC (no connected). Ricordiamo che osservando dall'alto il componente, la mezza luna nel caso d PDIP e il cerchietto nel caso di QFP e PLCC, individuano il pin numero 1, tutti gli altri seguono come da standard in senso antiorario.

Ogni pin del PIC presenta una specifica funzione. Più in generale è però sempre opportuno individuare gruppi di pin dal punto di vista funzionale. È un modo di approcciare la conoscenza del dispositivo del tutto generale che può essere applicato a qualunque PIC e, più in generale, a qualunque controllore. Quello che può cambiare da un modello all'altro è la specifica composizione di questi gruppi funzionali di pin e le specifiche elettriche degli stessi. Per questo è necessario, nel momento in cui si passa alla stesura dello schema elettrico del circuito a microcontrollore, disporre dello specifico datasheet.

Il primo gruppo di pin che occorre individuare è quello che ne consente alimentazione e il funzionamento dal punto di vista del clock e della programmazione. Nel caso del PIC16F877 in package PDIP 40 si tratta dei pin:

- VDD (alimentazione positiva +5 V) - pin 11 e 32;
- VSS (massa) – pin 16 e 31;
- MCLR (Master Clear) – pin1. Di norma va collegato al positivo dell'alimentazione; se riceve anche un breve impulso logico 0 genera il reset del controllore, il che comporta il riavvio del firmware in esecuzione. Si noti infatti, il simbolo di negato presente sull'acronimo del relativo pin **(figura 1)** che significa appunto che la funzione di clear è attiva con il pin portato a livello basso cioè a zero logico.
- OSC1 e OSC2- pin 13 e 14 rispettivamente. Per il momento ci limitiamo a segnalare che questi due pin servono per realizzare il circuito di oscillazione ovvero di generazione del clock del micro. Senza un clock che scandisca le operazioni di prelievo delle istruzioni dalla memoria e di esecuzione delle stesse, il controllore sarebbe infatti un oggetto con il "cervello congelato" incapace di reagire. Avremo modo di approfondire nel seguito del corso l'argomento.
- PGM (Low Voltage Programming Input) - pin 36, PGC (Programming Clock) - pin 39 e PGD (Programming Data) - pin 40, pin utili alla programmazione e al debugging in-circuit.

A tale proposito, occorre infatti osservare che il PIC può essere programmato inserendolo su di un apposito circuito detto programmatore. Questo è a sua volta generalmente interfacciato a un PC attraverso un cavo USB. Il programmatore utilizza proprio questi pin, oltre ai pin Vdd, Vss ed MCLR, per riversare il firmware, sotto forma di codice eseguibile, sulla memoria flash del controllore. Eseguita l'operazione, il controllore può essere

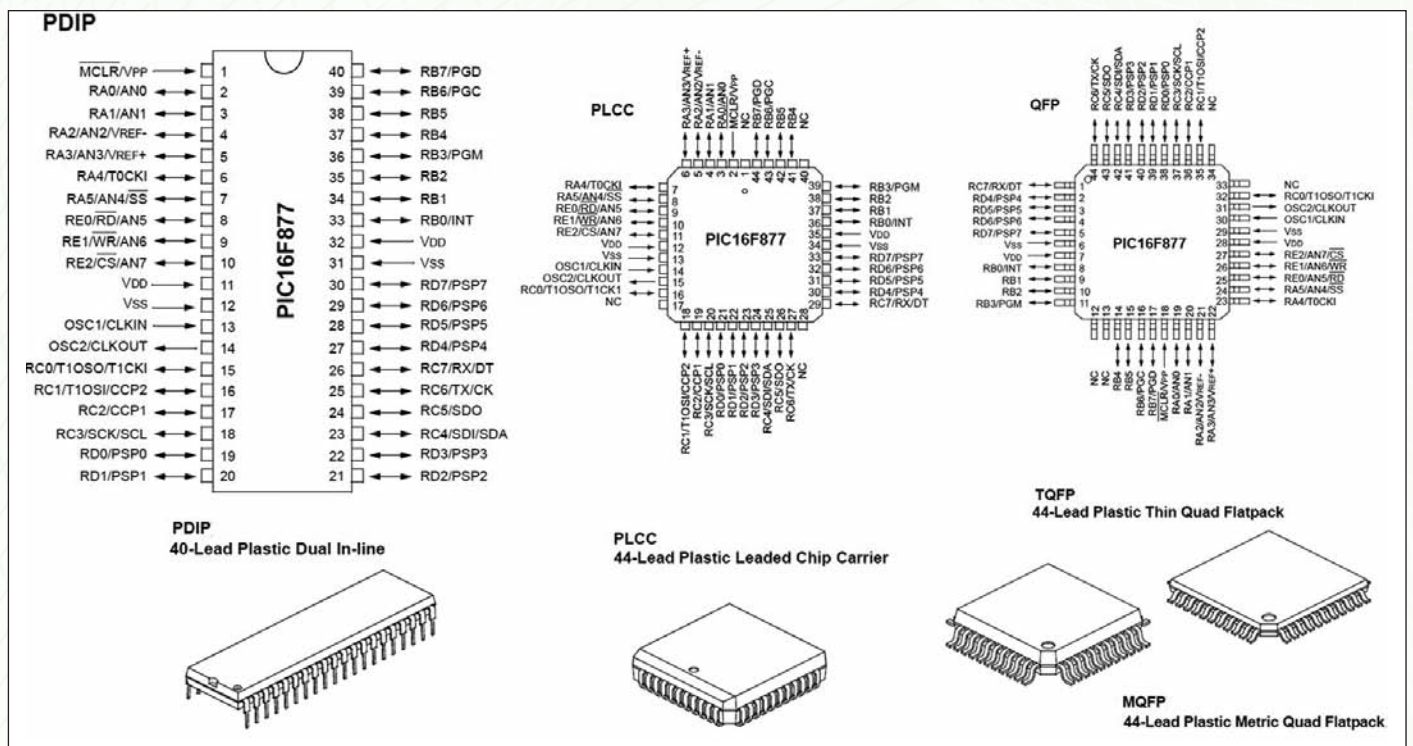
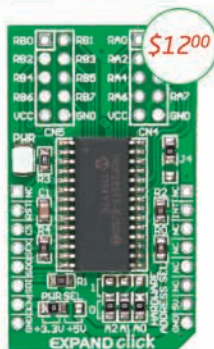
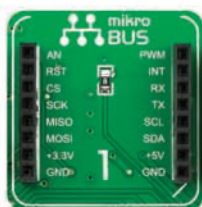
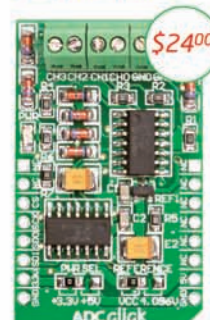
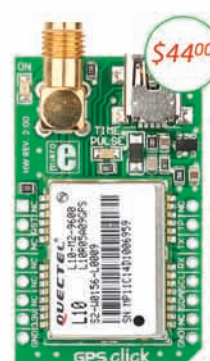
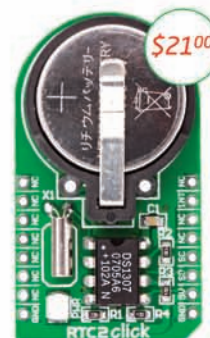
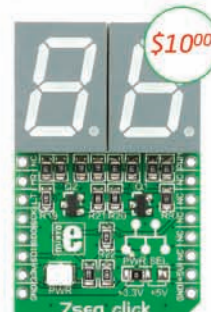
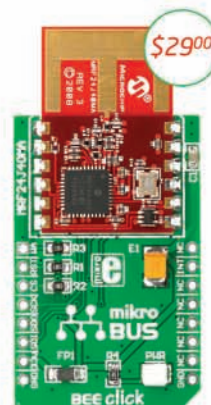
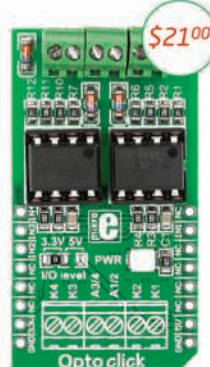



Figura 1: package e pin-out del controllore PIC16F877 (versione PDIP a 40 pin, PLCC e QFP a 44 pin).

rimosso dal programmatore e inserito sul circuito a microcontrollore per il quale era stato pensato. In alternativa è possibile portare opportunamente questi pin su un adeguato connettore sulla stessa scheda finale su cui il controllore verrà utilizzato. In questo modo il programmatore può essere collegato a questi pin senza bisogno di essere rimosso dalla sua board. Questo consente di eseguire una programmazione e al limite anche un debugger in circuit dell'applicazione. Le schede di sviluppo come le EasyPIC invece integrano generalmente on-board il programmatore.
I pin PGC, PGM e PGD coincidono anche, nel PIC16F877, rispettivamente con RB3, RB6 ed RB7. Il doppio acronimo indica semplicemente che questi pin svolgono una doppia funzione. Nella fattispecie è possibile utilizzare questi pin per la programmazione, ma durante l'esecuzione del firmware essi possono tranquillamente rispondere come pin di I/O.
La duplice (o triplice) funzionalità di alcuni pin (si parla più precisamente di funzioni multiplexate) è ricorrente per molti dei pin delle porta di I/O e viene gestita attraverso i bit dei registri di configurazione del controllore. Avremo modo di approfondire questo aspetto; per il momento, senza addentrarci oltre nelle funzionalità multiple dei pin di I/O, ci limitiamo ad osservare che completano il pin-out del 16F877:

- 6 pin della porta A (RA0…RA5);
- 8 pin della porta B (RB0…RB7);
- 8 pin della porta C (RC0…RC7);
- 8 pin della porta D (RD0…RD7);
- 3 pin della porta E (RE0…RE2);

Come al solito, nel caso in cui abbiate a che fare con un diverso modello di PIC, le porte disponibili possono risultare in numero maggiore o minore e il pin-out differire da quello del 16F877 che stiamo prendendo a riferimento ma la configurazione generale e l'approccio funzionale al chip non cambia.

PORTE I/O E IMPLICAZIONI ELETTRICHE SULLA PROGETTAZIONE DELL'APPLICAZIONE

Anche chi è inesperto della materia si aspetta che un controllore, in quanto entità capace di gestire un processo, sia in grado di acquisire ingressi e fornire o pilotare uscite secondo un determinato algoritmo impresso nel suo firmware e che prevede l'elaborazione degli stessi ingressi. Evidentemente, questo è il concetto fondamentale che dobbiamo imparare a padroneggiare da un punto di vista pratico se intendiamo utilizzare il controllore per realizzare delle applicazioni utili.
Chiaramente, ingressi e uscite possono essere livelli o stati logici, ma anche segnali analogici oppure dati comunicati secondo uno specifico protocollo dati basato a sua volta su uno specifica interfaccia o protocollo hardware, cioè fisico, in quanto basato su segnali elettrici ben precisi. L'argomento, descritto in questa maniera nella sua globalità, potrebbe apparire piuttosto complesso se non lo si affronta in maniera organica e graduale. Per questo motivo, in questa prima lezione dedicata alla pratica ci limiteremo a proporre e analizzare semplicemente alcune applicazioni in grado di pilotare lo stato logico dei pin di uscita del controllore. Attraverso essi potremo attivare circuiti esterni che noi ci limiteremo a simulare, senza ledere alcuna generalità, con semplici diodi LED.
Detto questo, analizziamo più da vicino alcuni aspetti elettrici propri del PIC 16F877. Abbiamo già detto dell'alimentazione richiesta pari a 5 Vdc. Da ciò deriva la possibilità di pilotare le porte di uscita e quindi i relativi pin via firmware, facendo assumere loro i valori 5 V e 0 V. Assodato questo, appare ovvio chiederci se sia possibile pilotare con questi livelli di tensione un circuito esterno. La risposta è ovviamente affermativa, a patto che il livello di tensione e la corrente che il circuito da pilotare richiede siano compatibili con quelli che il controllore è in grado di generare sui suoi pin di uscita.
Consideriamo allora il caso semplice in cui si intenda collegare direttamente un led su di un pin di uscita in modo da utilizzare livelli 5 V e 0 V per determinarne l'accensione o lo spegnimento. Evidentemente, dovremo per lo meno porre in serie al diodo una resistenza di limitazione. Senza di essa produrremmo un corto sul pin di uscita che danneggerebbe irreversibilmente il circuito di uscita del controllore che fa capo a quel pin. Se vogliamo far accendere il LED con una corrente tipica di 10-15 mA dovremo porre in serie una resistenza di circa (5 – Vd) / 12 mA = 330 ohm, dove Vd è la caduta sul diodo LED. Fortunatamente, il valore di resistenza ottenuto è un valore standard; nel caso in cui non lo fosse dovremmo scegliere una resistenza standard leggermente superiore o inferiore. Il problema può in realtà apparire banale ma non lo è affatto. Quando dobbiamo infatti pilotare un circuito esterno con i pin del controllore, non dobbiamo mai dare per scontato che esso sia in grado di fornire con i suoi pin di uscita la corrente richiesta dal circuito. Se si assorbe infatti, da un pin del controllore, una corrente superiore a quella che esso è in grado di fornire, quella porta prima o poi andrà incontro a un guasto. Bisogna allora valutare la capacità del PIC di erogare corrente e confrontarla con quella richiesta. Occorre per questo valutare i dati elettrici caratteristici del controllore normalmente riportati nei paragrafi intitolati "Electrical Characteristics-Absolute Maximum Ratings". In **tabella 2** riportiamo gli "Absolute Maximum Ratings" del PIC16F877, così come figurano sul datasheet del controllore **[1]**. Ovviamente, si tratta di dati che possono variare da modello a modello di controllore PIC. Come è possibile osservare dai dati riportati nella suddetta **tabella 2**, il singolo pin di I/O utilizzato come uscita può fornire al massimo 25 mA di corrente. Questo significa che utilizzarlo per fornire 12 mA a un LED non pone particolari problemi, tanto più se si intende pilotare lo stesso LED in maniera intermittente. Il discorso da fare non è diverso, qualunque sia il dispositivo che il pin o un insieme di pin devono pilotare. Ne consegue che, nel caso in cui la corrente o le correnti richieste siano maggiori rispetto a quelle espresse come Absolute Maximum Ratings, sarà necessario pilotare il dispositivo o i dispositivi di uscita interponendo tra loro e il controllore un circuito di interfaccia. Il discorso vale non solo per quanto riguarda i valori di corrente ma anche per quanto riguarda la compatibilità dei valori di tensione.
È importante osservare che, in generale, ammesso che vi sia compatibilità per quanto riguarda i livelli di tensione, è necessario:

1. non superare la massima corrente erogabile dal singolo pin di I/O;
2. non superare la massima corrente complessivamente erogabile da una porta di I/O che racchiude un certo numero di pin;
3. non superare una certa corrente massima assorbita da Vdd. Infatti, se si vede il controllore come un macronodo, la somma delle correnti che entrano nel dispositivo deve essere uguale alla somma delle correnti che vi escono, per cui se il pin Vdd non può assorbire più di una certa corrente, la somma di tutte le correnti fornite in uscita dalle singole porte sicuramente non dovrà superare lo stesso valore di corrente.

Stressare il dispositivo al di sopra delle sue reali possibilità può determinarne un danneggiamento permanente. Impiegare invece il dispositivo per lunghi periodi alle condizioni limite (Absolute Maximum Ratings appunto) può comportare una riduzione della sua affidabilità.

Le porte di I/O meritano in realtà ulteriori approfondimenti che contiamo di trattare nella prossima lezione. Quanto esposto in merito è per il momento più che sufficiente alla comprensione degli argomenti.

LA NOSTRA PRIMA APPLICAZIONE

Nei primi esperimenti pratici nell'uso di MikroPascal collegheremo uno o più diodi LED sulle uscite del PIC e li piloteremo in maniera tale da farli lampeggiare a una certa frequenza. Ovviamente, se utilizzassimo un PIC differente dal 16F877, con pochissime modifiche allo schema elettrico e quasi nessuna al firmware potremo ripetere l'esperimento senza particolari difficoltà. Più precisamente, tramite questi esempi gli esempi perseguiremo le finalità precedentemente elencate ai punti 1...9 del paragrafo introduttivo dell'articolo.

SCHEMA ELETTRICO

Lo schema elettrico che implementa l'hardware delle applicazioni di esempio che stiamo per mostrare è riportato in **figura 2**. Le resistenze di limitazione sono state scelte in maniera tale da poter osservare il lampeggio dei LED senza assorbire mai dal singolo pin, dalla singola porta e dall'intero controllore le correnti massime consentite nel rispetto dei Maximum Absolute Ratings riportati in **tabella 2**. Ciò non toglie la possibilità, qualora lo sperimentatore lo desideri, di semplificare lo schema sperimentando il comportamento di alcune sole uscite del controllore, in modo da limitare il numero di LED e la complessità dell'implementazione.

È in ogni caso opportuno notare:
1. la resistenza di pull-up sull'uscita RA4;
2. il connettore (JP) ed i jumper (J1 e J2) per la programmazione in-circuit del controllore attraverso programmatori quali l'ICD2;
3. il pulsante di reset;
4. l'alimentazione Vdd (5V) e Vss (GND);
5. la connessione a Vdd del pin MCLR;
6. la sezione di oscillazione per la generazione del clock (Quarzo X1 e condensatori C1 e C2 che insistono sui pin OSC1 ed OSC2).

Nelle prossime puntate non mancheremo di spendere qualche parola in più sulle funzioni legate a questi componenti e al loro coretto dimensionamento. Per il momento evitiamo specifici approfondimenti.

Il circuito può essere riprodotto senza particolari difficoltà con una qualunque scheda di sviluppo (es. EasyPIC) mentre la programmazione della memoria flash può avvenire attraverso l'eventuale programmatore integrato on-board oppure attraverso programmatori quali ICD2 ed ICD3. Anche questo aspetto sarà trattato con maggiore enfasi in altri frangenti del corso.

CREAZIONE DEL FIRMWARE E PRIMI ESPERIMENTI CON EDITOR E COMPILATORE

Per scrivere il firmware che consente di pilotare le uscite del controllore, come utilizzate secondo lo schema di **figura 2**, avviamo prima l'ambiente di sviluppo MikroPascal e procediamo poi alla creazione di un nuovo progetto attraverso il comando New Project disponibile all'interno della Start Page **(figura 3)**.

Il form di dialogo che si apre (New Project Wizard-Figura 4) notifica all'utente che si è avviata una procedura guidata che consente rispettivamente:
- la creazione del progetto;
- la selezione del modello di PIC oggetto dell'applicazione firmware che si intende scrivere (Target Device);
- l'impostazione della frequenza di clock dell'applicazione;
- la definizione dei file e delle librerie che comporranno il progetto.

Procediamo con Next **(figura 4)** allo Step 1 ed impostiamo rispettivamente, nel nuovo form che compare in successione **(figura 5)**, il nome del progetto (Project Name; nell'esempio proposto abbiamo dato al progetto il nome "Pilotaggio LED"), la cartella di archiviazione dello stesso (Project Folder), selezioniamo il controllore oggetto della nostra applicazione (Device Name; PIC16F877 nel nostro caso) e la frequenza di clock (Device clock; 8 MHz nel nostro caso). A riguardo, occorre sottolineare che è necessario impostare il clock in maniera tale che esso corrisponda alla frequenza del quarzo impiegato nella sezione fisica di clock del controllore (8 MHz, vedi schema elettrico di **figura 2**).

Procediamo a questo punto, con Next **(figura 5)**, all'accesso al prossimo form e quindi al prossimo step (Step 2, **figura 6**). Per il momento procediamo semplice-

TABELLA 2

CARATTERISTICA ELETTRICA DEL PIN (O DELLA PORTA) DI I/O	VALORE
Maximum current out of VSS pin	300 mA
Maximum current into VDD pin	250 mA
Maximum output current sunk by any I/O pin	25 mA
Maximum output current sourced by any I/O pin	25 mA
Maximum current sourced by PORTA, PORTB, e PORTE insieme	200 mA
Maximum current sourced by PORTC and PORTD insieme	200 mA

Tabella 2: caratteristiche elettriche delle porte di I/O del PIC 16F877 (Absolute Maximum Ratings) [1].

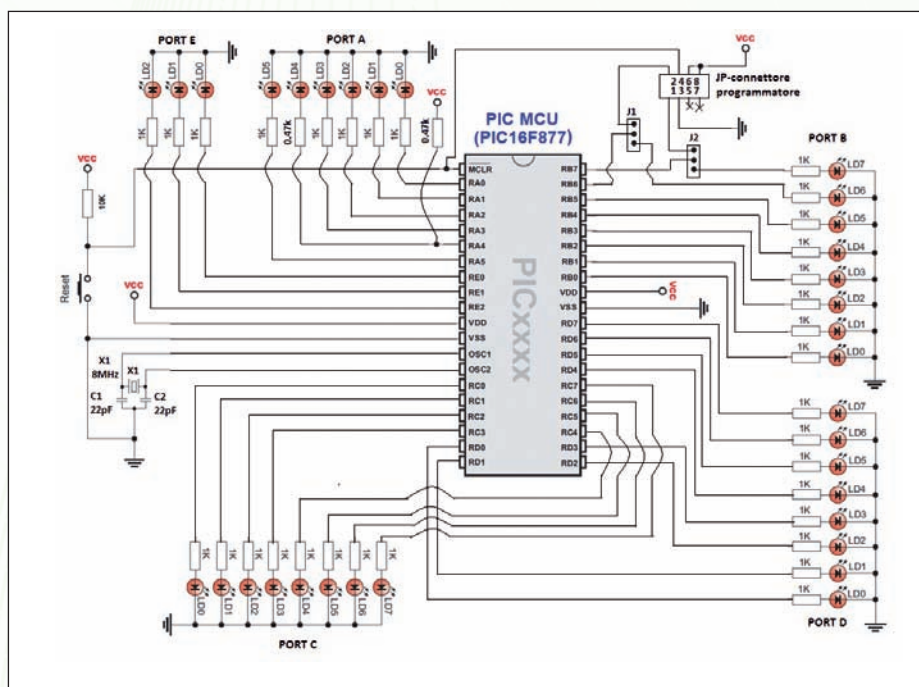



Figura 2: schema elettrico di riferimento per i firmware di esempio riportati rispettivamente nei listati Lampeggio Led-ver. a, b, c, d, e, f e g (vedi figure 17, 19, 20, 21, 22, 23, 24).

mente con Next (Step 2, **figura 6**) sul form e quindi allo step ancora successivo (Step 3, **figura 7**). Attraverso questo form è possibile includere nel progetto alcune librerie. Il principale vantaggio offerto dalle librerie integrate in MikroPascal risiede nel fatto di disporre di oltre 500 funzioni già scritte che possono essere inserite nei propri progetti e facilmente analizzate attraverso il Code Assitant (Ctrl+Spazio). La disponibilità di un'ampia gamma di funzioni di libreria, eventualmente organizzate per tipologia, è infatti uno dei punti forza di qualunque compilatore e quelli pensati per lo sviluppo di applicazioni firmware a microcontrollore da questo punto di vista non fanno eccezione. Procediamo ancora avanti con Next al prossimo step (Step 4, **figura 8**) poiché non abbiamo per il momento bisogno di inserire librerie. Nel nuovo form che compare **(figura 8)**, spuntiamo l'opzione "Open Edit Project window to set Configuration bit" e procediamo con Finish, operazioni che ci consentono di accedere all'impostazione dei bit di configurazione (Edit Project, **figura 9**).

Si tratta di un form specializzato per la definizione dei bit di configurazione e della frequenza di clock **(figura 9)**. Sullo sfondo appare l'editor con la struttura del codice Pascal preimpostata ma vuota di istruzioni.

Lasciamo per il momento le impostazioni dei bit di configurazione come mostrate in **figura 9** e procediamo oltre con OK. Il Wizard restituisce il controllo dell'editor. Al suo interno si nota, in particolare, il nome del programma (in alto) che corrisponde a ciò che abbiamo precedentemente impostato nello step di **figura 5**, la sezione destinata alla dichiarazione delle variabili (Declarations section), l'area destinata al corpo del programma (Main Program) racchiusa, come è tipico della struttura di un codice Pascal, tra le parole riservate "begin" e "end." (si noti in particolare il ";" alla fine del nome del programma e il "." dopo l'end). Giunti a questo punto, la procedura di impostazione e configurazione del progetto è davvero terminata. Nella cartella di progetto (Project Folder) definita al passo 1 **(figura 5)** compaiono già i file che compongono il progetto **(figura 10)**.

A questo punto possiamo iniziare la stesura del codice. Al fine di prendere dimestichezza con il linguaggio ed evitare errori banali nel proseguo è possibile eseguire alcuni esperimenti di compilazione e di sintassi già a questo punto anche se non abbiamo ancora scritto nemmeno una parola di codice. Quello che infatti è disponibile all'interno dell'area di editor è a tutti gli effetti un listato di codice MikroPascal, sebbene ancora privo di istruzioni. Per questo motivo può pertanto a tutti gli effetti essere sottoposto a compilazione.

Per eseguire la compilazione utilizziamo il comando Build dell'omonimo menu a tendina oppure la combinazione di tasti Ctrl+F9 **(figura 11)**. La compilazione del

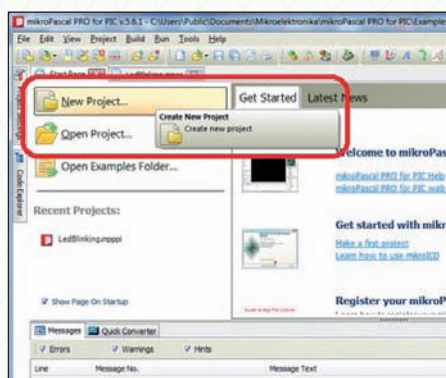



Figura 3: avvio della creazione di un nuovo progetto attraverso il comando "New Project" reso disponibile dalla Start Page.

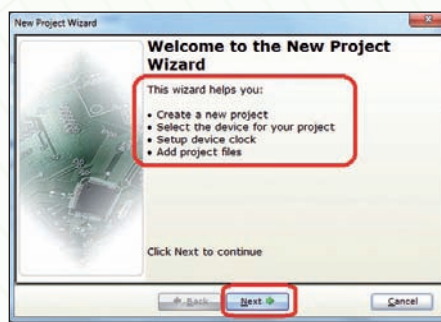



Figura 4: il New Project Wizard avviato evidenzia i passi che saranno percorsi con la procedura guidata di creazione del nuovo progetto. Attraverso il pulsante "Next" è possibile passare al primo di questi step.

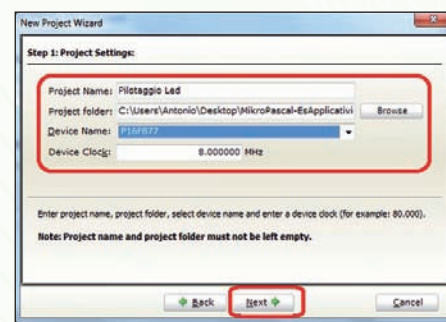



Figura 5: New Project Wizard, Step1 - Project Settings (Impostazione del nome del progetto, della cartella di progetto, del target device e della frequenza di clock).

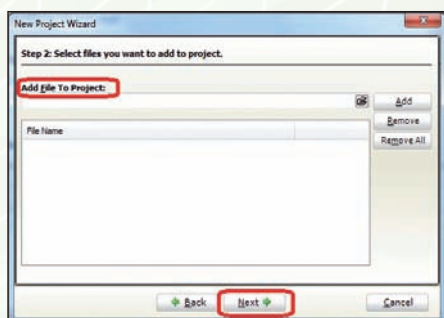



Figura 6: New Project Wizard, Step 2 - selezione dei file da aggiungere al progetto.

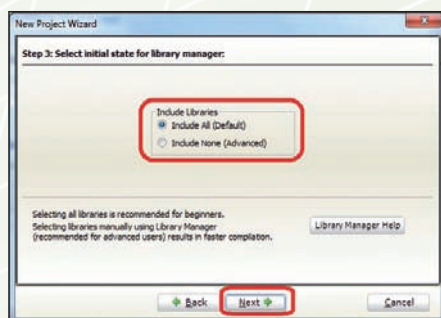



Figura 7: New Project Wizard, Step 3 - inclusione di librerie.

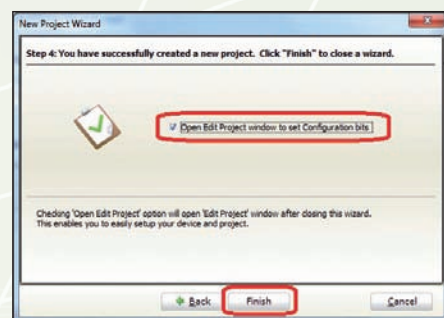



Figura 8: New Project Wizard, Step 4 - fine della procedura guidata e accesso alla pagina Configuration bit (spunta della voce "Open Edit Project window…").

codice MikroPascal produce come output il codice eseguibile .hex. A compilazione avvenuta, nella sezione Messages (se non è presente nella vostra schermata richiamatela attraverso il comando Messages del menu a tendina View) compare il log di tutte le operazioni eseguite in fase di compilazione e l'esito di ciascuna di esse. È opportuno dare un'occhiata a questi dati al fine di renderci effettivamente conto del buon esito delle operazioni. Dalla lettura dei messaggi di log **(figura 12)** è possibile osservare che:

1. la compilazione è realmente avvenuta sul progetto contenuto nella cartella di progetto impostata;
2. il controllore per il quale è avvenuta la compilazione è effettivamente quello da noi selezionato precedentemente (PIC16F877). Di questo è indicato anche la memoria RAM e ROM disponibile;
3. è notificato l'esito della compilazione e del linking (il fatto di non avere al momento funzioni di libreria inserite non toglie nulla al processo generale di compilazione e linking);
4. è notificata la quantità di RAM e di ROM utilizzata e quella libera; in questo caso la quantità di ROM utilizzata è quasi nulla, essendo il programma ancora privo di istruzioni, ma in altri casi la richiesta di una quantità di memoria superiore a quella fisicamente disponibile porterebbe a un errore che farebbe fallire il processo di compilazione e linking e che verrebbe notificata come errore tra i messaggi. Gli errori sono notificati nell'area Messages in rosso, in maniera da attirare l'attenzione dello sviluppatore. Avremo modo di approfondire in altre occasioni questo aspetto.

L'analisi dei messaggi di log è comunque particolarmente importante in quanto aiuta lo sviluppatore a comprendere quali siano le cause che hanno portato al fallimento del processo di compilazione spingendolo ad apportare prontamente le necessarie correzioni. Ovviamente dobbiamo attenderci, nel momento i cui carichiamo questo eseguibile sul PIC, che lo stesso giri a vuoto dal momento che non contiene istruzioni da eseguire.

Il processo di compilazione e linking produce nuovi file che ritroviamo all'interno della cartella di progetto. Tra questi troviamo, in particolare, il file eseguibile .hex che va caricato nella memoria flash del controllore per l'esecuzione **(figura 13)**.

Possiamo già fare alcuni esperimenti di sintassi simulando volutamente degli errori. Proviamo, per esempio, a togliere una lettera qualunque alla parola riservata program e a ricompilare. Il risultato è riportato in **figura 14**. Il compilatore si aspetta di trovare come inizio programma la parola riservata "program". Non trovandola interrompe la compilazione e segnala l'errore di sintassi "Syntax Error: Expected "program" but "program" found"… "Finished with error".

Proviamo ora a modificare le parole chiavi program, begin ed end sostituendo ad alcune lettere il maiuscolo. È possibile verificare che in compilazione non viene generato alcun errore o warning. Questo

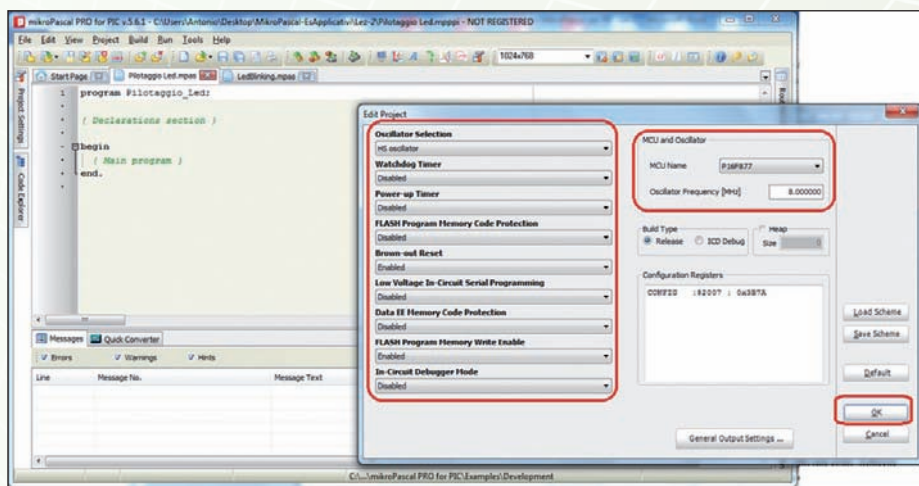



Figura 9: Edit Project Window (impostazione dei bit dei registri di configurazione).

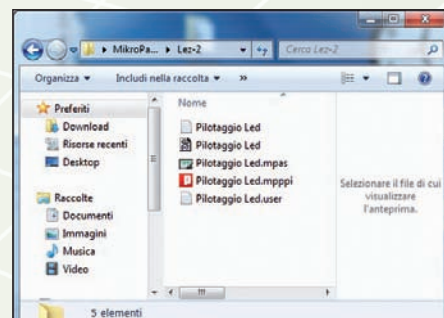



Figura 10: i file di progetto nella cartella di progetto (Project Folder).

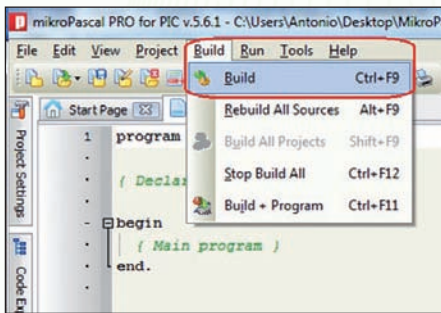



Figura 11: compilazione del progetto per la generazione del file .hex da caricare sulla flash memory del PIC (Comando Build dell'omonimo menu a tendina o combinazione Ctrl+F9).

comportamento non è anomalo ed è motivato dal fatto che il Pascal è un linguaggio case insensitive, nel senso che non fa distinzione tra maiuscole e minuscole.
Se invece proviamo a togliere il ";" dopo il nome del programma nella prima linea di programma oppure a togliere l'end o il solo "." dopo di esso vediamo che la compilazione genera una segnalazione di errore. Il compilatore capisce infatti che il nome del programma è terminato proprio dal ";" e che il programma è terminato proprio dall'"end". Non trovando questi indicatori durante la fase di compilazione, il compilatore capisce che nel listato di programma vi sono anomalie e le segnala come errori.
In **figura 15** è riportata la struttura generale del listato di programma con la sezione di dichiarazione delle variabili (Declarations section) e il corpo del programma (Main Program) ancora da riempire. Al fine di non incorrere in equivoci, si fa osservare che tutto quanto è compreso tra parentesi graffe è inteso dal compilatore come commento e pertanto ignorato in fase di compilazione. Il commento è inoltre caratterizzato dal classico colore verde che lo contraddistingue. È inoltre interessante osservare come, qualora il listato di programma presenti una differenza rispetto all'ultima versione salvata, anche per la sola eliminazione o aggiunta di uno spazio, l'editor fa comparire automaticamente una linea verticale di colore giallo all'estrema sinistra del listato stesso in corrispondenza della modifica. È sufficiente eseguire un "Salva Progetto" (Save Project As… del menu Project, **figura 16**) per far scomparire questa notifica. Si tratta di una segnalazione particolarmente utile in quanto aiuta il programmatore ad avere il controllo di quanto è stato modificato dopo l'ultima operazione di salvataggio del progetto. Contribuisce al controllo del listato

Figura 12: area Messages (notifica delle operazioni di compilazione e linking).

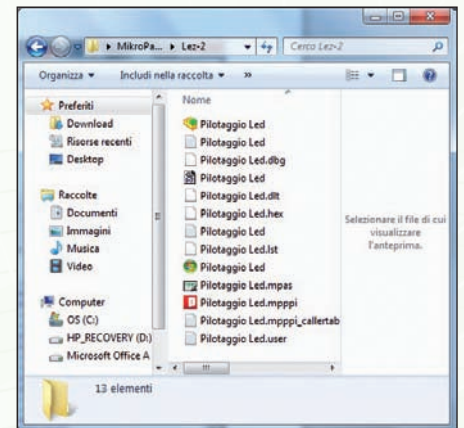



Figura 13: file contenuti nella Ccartella di progetto dopo l'avvenuta compilazione.

anche la numerazione delle righe di codice che l'editor mostra automaticamente lungo il bordo sisnistro della schermata **(figura 15)**.
Spendiamo qualche parola a riguardo dell'inserimento di commenti all'interno del codice. Qualora si debbano inserire diverse righe di commento consecutive in un punto del codice è opportuno porre all'inizio di questo una parentesi graffa aperta { e, alla fine, una parentesi graffa chiusa }. Il compilatore capisce che tutto quanto è compreso tra questi due caratteri riservati è un commento e lo può in questo modo ignorare in fase di compilazione. Qualora invece il commento occupi una sola riga del listato è sufficiente far precedere lo stesso da //. I commenti utilizzati correttamente, secondo quanto appena esposto, vengono automaticamente colorati in verde dall'editor e sono in tal modo facilmente individuabili.
Si tenga presente che l'inserimento di commenti in testa al listato o in prossimità di istruzioni, funzioni, blocchi o strutture di controllo specifiche può aiutare moltissimo la corretta comprensione del codice sia durante il suo sviluppo sia in occasione di operazioni di manutenzione eseguite anche da parte di terzi programmatori, a tutto beneficio del time-to-market del progetto e della sua manutenibilità. Un impiego corretto ed efficace dei commenti è per questo una prassi estremamente importante che occorre far propria fin da subito. Nel tempo risulterà sempre più utile e indispensabile tanto più quanto più complessi diverranno le applicazioni che svilupperemo.

CONFIGURAZIONE E PILOTAGGIO DELLE PORTE DI USCITA

Analizziamo a questo punto, con riferimento allo schema elettrico di **figura 2**, una serie di listati MikroPascal, con lo scopo di mostrare come utilizzare i pin del PIC come uscite e come pilotare tramite di esse il lampeggio di uno o più LED a frequenza prefissata. Ciascuno dei listati mostrati a partire dalla **figura 17** evidenzia un aspetto pratico specifico sull'uso di MikroPascal nel controllo delle porte di uscita del PIC. Come avremo modo di sperimentare nelle prossime puntate, tali porte sono più in generale porte di I/O e non semplicemente di uscita. Per il momento ci limiteremo tuttavia ad utilizzarle come uscite. Per far ciò sarà necessario utilizzare apposite istruzioni che agiscono sui registri di configurazione delle porte in modo da configurarle opportunamente.

Listato 1: lampeggio Led–ver.a

Il programma, il cui listato è riportato in **figura 17**, ha come scopo quello i mostrare come configurare i registri di direzione delle porte di I/O in modo da poterle utilizzare come uscite e come pilotare le uscita di ciascuna porta. Data la sua semplicità è paragonabile al classico "Hello World".
Con la commutazione ON/OFF dei pin corrispondenti ai bit di ciascuna porta si genera l'accensione e lo spegnimento alternato dei led connessi sugli stessi pin.
Con l'istruzione
TRISA:=0;

Figura 15: struttura del codice MikroPascal (si noti il nome del programma, la sezione di dichiarazione delle variabili e quella del Main Program. Si notino inoltre le parole chiavi program, begin ed end).

si assegna a ciascuno dei bit del registro TRISA del controllore valore logico 0. TRISA è un registro a 8 bit del controllore che serve a definire la configurazione di ciascuno dei pin della porta A, cioè la direzione di funzionamento (ingresso/uscita). A ciascun bit del registro corrisponde un pin della corrispondente porta. Se quel bit del registro assume valore "1" il pin è configurato come ingresso, se invece assume valore "0" il pin è configurato come uscita. L'istruzione di assegnazione TRISA:=0 (:= è l'operatore di assegnazione) esegue l'assegnazione dello zero decimale al registro TRISA. Poiché lo zero decimale corrisponde in binario alla sequenza di 8 bit 00000000, si sono in questo modo configurati, in un solo colpo, tutti i pin della porta A come uscite. Analogamente, l'istruzione
PORTA:= 0x00;
consente di assegnare, in un solo colpo, a tutti i bit della porta A il valore logico 0 (da cui lo spegnimento di tutti i LED che insistono sui pin della porta A), mentre l'istruzione
PORTA:= 0xFF;
determina analogamente l'assegnazione del valore logico 1 agli stessi bit, da cui l'accensione di tutti i suddetti LED. 0x indica

Figura 16: salvataggio del progetto (Save Project As... del menu a tendina Project).

Figura 17: listato programma "Lampeggio Led-ver.a". Si noti la struttura del listato e le parole chiave, l'inserimento dei commenti, la configurazione delle porte, l'assegnazione dei valori logici ai bit delle singole porte configurate come uscite e l'impiego della funzione Delay_ms(1000) impiegata per la generazione di un ritardo di 1000 millisecondi.

infatti che il valore che segue è un esadecimale per cui 0xFF= (11111111) binario. Ciascun bit esprime il valore che si sta assegnando al corrispondente bit della porta in questione (porta A in questo caso). Cosa analoga dicasi per le istruzioni che pilotano tutte le altre porte. Si rimanda al listato del programma ver. a riportato in **figura 17** per l'osservazione di quanto detto.
Qualche parola merita la funzione
Delay_ms(1000);
che consente di ottenere un ritardo di 1.000 millisecodi = 1 secondo da cui il lampeggio dei LED a frequenza di 0,5 Hz (si noti come in ogni ciclo While siano contenuti 2 istruzioni Delay_ms). De-

Figura 14: notifca di messaggi di errore durante la compilazione.

Figura 18: è possibile verificare e/o modificare le impostazioni del target device e della frequenza di clock attraverso Project Setting.

```
program Pilotaggio_Led;
{ Dichiarazione variabili }
begin
  TRISA := 0;         // imposta l'intera porta A come output
  TRISB := 0;         // imposta l'intera porta B come output
  TRISC := 0;         // imposta l'intera porta C come output
  TRISD := 0;         // imposta l'intera porta D come output
  TRISE := 0;         // imposta l'intera porta E come output

  while TRUE do
    begin
      PORTA := $00;   // Spegne i LED sulla porta A
      PORTB := $00;   // Spegne i LED sulla porta B
      PORTC := $00;   // Spegne i LED sulla porta C
      PORTD := $00;   // Spegne i LED sulla porta D
      PORTE := $00;   // Spegne i LED sulla porta E
                      // $00 (notaz. esadecimale)=00000000 (notaz. binaria)
      Delay_ms(1000); // 1 secondo di ritardo

      PORTA := $FF;   // Accende i LED sulla porta A
      PORTB := $FF;   // Accende i LED sulla porta B
      PORTC := $FF;   // Accende i LED sulla porta C
      PORTD := $FF;   // Accende i LED sulla porta C
      PORTE := $FF;   // Accende i LED sulla porta E
                      // $FF (notaz. esadecimale)=11111111 (notaz. binaria)
      Delay_ms(1000); // 1 secondo di ritardo. Delay_ms(x) significa x
                      // millisecondi di ritardo. Senza di esso l'occhio umano
                      // non riuscirebbe a percepire il lampeggio del led
    end;              // End del Main program
end.
```

Figura 19: listato di programma "Pilotaggio Led ver.b". La notazione esadecimale caratterizzata dal prefisso 0x può essere sostituita da quella con prefisso $. A parte questo aspetto, il codice di programma è del tutto equivalente a quello della versione a mostrato in figura 17.

```
program Pilotaggio_Led;
{ Dichiarazione variabili }
begin
  TRISA := 0;         // imposta l'intera porte A come output
  TRISB := 0;         // imposta l'intera porta B come output
  TRISC := 0;         // imposta l'intera porta C come output
  TRISD := 0;         // imposta l'intera porta D come output
  TRISE := 0;         // imposta l'intera porta E come output
                      // 0 decimale corrisponde a 00000000 binario. Tutti i
                      // bit di ciascun registro di direzione TRISx delle
  while TRUE do       // porte x assunono pertanto valore 0 configurndo il
    begin             // relativo pin come uscita
      PORTA := 0;     // Spegne i LED sulla porte A
      PORTB := 0;     // Spegne i LED sulla porte B
      PORTC := 0;     // Spegne i LED sulla porta C
      PORTD := 0;     // Spegne i LED sulla porta D
      PORTE := 0;     // Spegne i LED sulla porta E
                      // 0 decimale corrisponde a 00000000 binario. Tutti i
                      // bit di ciascuna porta di uscita assunomo valore 0
                      // da cui lo spegnimento dei led
      Delay_ms(1000); // 1 secondo di rtardo
      PORTA := 255;   // Accende i LED sulla porta A
      PORTB := 255;   // Accende i LED sulla porta B
      PORTC := 255;   // Accende i LED sulla porta C
      PORTD := 255;   // Accende i LED sulla porta C
      PORTE := 255;   // Accende i LED sulla porta E
                      // 255 decimale corrisponde a 11111111 binario. Tutti i
                      // bit di ciascuna porta di uscita assunomo valore 1
                      // da cui l'accensione dei led
      Delay_ms(1000); // 1 secondo di ritardo. Delay_ms(x) significa ritardo
                      // di x millisecondi. Senza di esso l'occhio umano non
                      // sarebbe in grado di percepire il lampeggio
    end;              // End Main program
end.
```

Figura 20: pilotaggio Led ver.c. Si noti come al posto della notazione esadecimale possa essere utilizzata la più familiare notazione decimale. A parte questo dettaglio, il codice è del tutto simile a quello delle revisioni a e b.

lay_ms sta per "ritardo espresso in millisecondi". Quando l'esecuzione del codice giunge in questo punto, il processore non fa altro che attendere il trascorrere del tempo indicato come argomento della funzione (espresso in millisecondi). Il lampeggio è dovuto al ciclo principale While TRUE che dopo le istruzioni di configurazione delle porte viene eseguito indefinitamente determinando il susseguirsi rispettivamente di un blocco di istruzioni che assegnano bit 0 alle uscite, di un ritardo di un secondo e di un blocco di istruzioni che assegnano valore 1 alle stesse uscite. Variando il valore assegnato come argomento alla funzione è possibile ottenere una frequenza di lampeggio differente. Si noti come nel listato non compaiano direttive particolari per il compilatore dal momento che sia il target device (modello di controllore sulla cui memoria riversare il firmware) che la frequenza di clock sono già stati impostati attraverso il percorso guidato di creazione del nuovo progetto, come risulta nella pagina Project Setting **(figura 18)**.

Altri punti sui quali è opportuno soffermare l'attenzione sono:

- l'operatore di assegnazione :=;
- il corpo del programma compreso tra le parole chiavi begin ed end (non conta se scritto in minuscolo o maiuscolo essendo il Pascal case insensitive);
- il ciclo while che si chiude con un end; che ha significato "esegui fino a che…" e che avremo modo di analizzare più in profondità, come altre strutture di controllo, durante lo sviluppo del corso;
- la parola chiave end. alla fine del programma;
- il ; alla fine di ciascuna istruzione;
- il nome del programma preceduto dalla parola chiave program e terminato ancora una volta dal ;.

Tutti questi sono tratti propri della sintassi del Pascal che occorrerà tenere bene a mente d'ora in avanti, sia durante la scrittura del codice sia in occasione della notifica di errori di compilazione cui si dovrà porre rimedio.

È importante osservare che l'intero programma può essere composto, in generale, da più blocchi Begin End appartenenti a diverse strutture di controllo annidate. Ciascuno di questi blocchi è evidenziato dall'editor attraverso linee punteggiate rosse sul lato sinistro dell'editor stesso **(figura 17)**. Anche questo accorgimento facilita l'individuazione di blocchi di codice e il riconoscimento di strutture di controllo annidate, aiutando il programmatore a mantenere costantemente

```
program Pilotaggio_Led;
{ Dichiarazione variabili }
begin
  TRISA := 0x1F; //0x1F=00011111 => RA0,RA1,RA2,RA3,RA4 ingressi, RA5 uscita
  TRISB := 0x0F; //0x0F=00001111 => RB0,RB1,RB2,RB3 ingressi; RB4,RB5,RB6,RB7 us
  TRISC := 0x0F; //RC0,RC1,RC2,RC3 ingressi; RC4,RC5,RC6,RC7 uscite
  TRISD := 0x0F; //RD0,RD1,RD2,RD3 ingressi; RD4,RD5,RD6,RD7 uscite
  TRISE := 0x0F; //RE0,RE1,RE2 ingressi

  while TRUE do
    begin
      PORTA := 0;     // Spegne i LED sui pin RA5
      PORTB := 0;     // Spegne i LED sui pin RB4,RB5,RB6,RB7
      PORTC := 0;     // Spegne i LED sui pin RC4,RC5,RC6,RC7
      PORTD := 0;     // Spegne i LED sui pin RD4,RD5,RD6,RD7
      PORTE := 0;     // Non ha alcun effetto

      Delay_ms(1000); // 1 secondo di ritardo
                      // 255 decimale=11111111 binario ==>
      PORTA := 255;   // Accende i LED sui pin RA5
      PORTB := 255;   // Accende i LED sui pin RB4,RB5,RB6,RB7
      PORTC := 255;   // Accende i LED sui pin RC4,RC5,RC6,RC7
      PORTD := 255;   // Accende i LED sui pin RD4,RD5,RD6,RD7
      PORTE := 255;   // Non ha alcun effetto
                      // tutti gli altri pin e quindi irelativi led sono spenti
                      // ed inattivi essendo configurati come ingressi
      Delay_ms(1000); // 1 secondo di ritardo
    end;              // End Main program
end.
```

Figura 21: pilotaggio Led ver.d. Questo esempio consente di chiarire ulteriormente quale sia la corrispondenza tra bit del registro di direzione di una porta TRISx, bit della stessa porta X e pin di I/O.

```
program Pilotaggio_Led;
{ Dichiarazione variabili }
begin
  TRISA := 0x00;          // imposta l'intera porta A come output
  TRISB := 0x00;          // imposta l'intera porta B come output
  TRISC := 0x00;          // imposta l'intera porta C come output
  TRISD := 0x00;          // imposta l'intera porta D come output
  TRISE := 0x00;          // imposta l'intera porta E come output

  while TRUE do
    begin
      PORTA := %00110011; // PORTA=%00110011 => RA0=RA1=1, RA2=RA3=0, RA4=RA5=1
                          // gli ultii due bit non producono effetti essendo
                          // la porta A estesa da RA0 ad RA5. Analogamente
      PORTB := %00110011; // ==> RB0=RB1=1, RB2=RB3=0, RB4=RB5=1, RB6=RB7=0
      PORTC := %00110011; // ==> RC0=RC1=1, RC2=RC3=0, RC4=RC5=1, RC6=RC7=0
      PORTD := %00110011; // ==> RD0=RD1=1, RD2=RD3=0, RD4=RD5=1, RD6=RD7=0
      PORTE := %00110011; // ==> RE0=RE1=1, RE2=0. Gli altri bit non hanno
                          //     effetto essendo la porta E estesa fino ad RE2
      Delay_ms(1000);     // 1 secondo di ritardo

      PORTA := %11001111; // Queste istruzioni invertono lo stato logico
      PORTB := %11001111; // delle porte rispetto ai valori impostati nel
      PORTC := %11001111; // precedente blocco da cui l'effetto di lampeggio
      PORTD := %11001111; // tranne che per i led RA0, RA1, RB0, RB1, RC0, RC1,
      PORTE := %11001111; // RD0, RD1, RE0 ed RE1 che rimangono sempre accesi
      Delay_ms(1000);     // 1 secondo di ritardo
    end;                  // End Main program
end.
```

Figura 22: lampeggio Led ver.e. Il programma è analogo alle precedenti versioni a, b, c e d con l'unica differenza di utilizzare la notazione binaria (prefisso %) per l'assegnazione del valore ai bit delle porte di uscita.

```
{
 * Corso: MikroPascal per PIC-Fare Elettronica-Lezione n.: 2
 * Nome progetto: Lampeggio Led -ver. f - Revisione: 2012-03-08
 * Autore: A. Giannico
 * Descrizione:
     Il programma è leggermente differente dai precedenti ver. a,b,c,d,e
     ed ha come scopo quello di mostrare come sia possibile eseguire un
     accesso diretto al singolo bit di un registro di direzione di una porta
     o della porta stessa.
 *   MCU:PIC16F877
     Scheda di sviluppo:Easypic3
     Oscillatore: HS 8.0000 MHz Crystal
     SW: mikroPascal PRO for PIC  ver. 5.61
     http://www.mikroe.com/eng/products/view/10/mikropascal-pro-for-pic/
  }
program Pilotaggio_Led;
{ Dichiarazione variabili }
begin
  TRISA.0 := 0;       // TRISA.0 significa bit 0 di TRISA==>imposta A0 come output
  TRISA.1 := 0;       // TRISA.1 significa bit 1 di TRISA==>imposta A1 come output
  TRISA.2 := 1;       // TRISA.2 significa bit 2 di TRISA==>imposta A2 come input
  while TRUE do
    begin
      PORTA.0 := 0;   // PORTA.0 significa bit 0 della porta A==> pin A0=0
      PORTA.1 := 0;   // PORTA.1 significa bit 1 della porta A==> pin A1=0
      PORTA.2 := 0;   // Spegnerebbe il LED sul pin A2 che tuttavia
                      // non può essere pilotato poiché TRISA.2=1
      Delay_ms(1000);// 1 secondo di ritardo

      PORTA.0 := 1;   // PORTA.0 significa bit 0 della porta A==> pin A0=1
```

Figura 23: lampeggio Led ver.f. Il programma è analogo alle precedenti versioni a, b, c e d con l'unica differenza di utilizzare l'accesso ai singoli bit del registro di configurazione TRISA o della porta PORTA.

sotto controllo l'evoluzione del codice e facilitando la ricerca e l'individuazione di errori di sintassi.

Listato 2: lampeggio Led–ver.b

Il programma, il cui listato è mostrato in **figura 19** è in tutto e per tutto identico alla ver.a, con l'unica differenza di impiegare, per l'assegnazione dei valori ai bit delle porte, la notazione esadecimale espressa con il simbolo $ al posto di quella con il prefisso 0x. Il programma esegue le stesse operazioni essendo le due notazioni perfettamente equivalenti (es 0x1F è perfettamente equivalente a $1F).

Listato 3: lampeggio Led–ver.c

Il programma, il cui listato è riportato in **figura 20**, è identico alle versioni a e b e mette in evidenza come i valori assegnati alle porte di uscita possano essere espressi anche in notazione decimale. Tutte le assegnazioni che vengono fatte ai registri di direzione TRISx e alle porte corrispondenti avvengono in questa versione del programma in notazione decimale. Si noti infatti che 255 (notazione decimale) corrisponde in binario alla sequenza 11111111 e, in esadecimale, ad FF.

Listato 4: lampeggio Led–ver.d

Ricordiamo che se un bit del registro TRISx di direzione è pari a 1, allora il bit che corrispondentemente assume la stessa posizione nella porta X e un ingresso. Al contrario, se un bit del registro TRISx di direzione è pari a 0, allora il bit che corrispondentemente assume la stessa posizione nella porta X è un'uscita. Nel listato del codice ver.d mostrato in **figura 21**, ai bit del registro di configurazione TRISA della porta A viene assegnato valore 0x1F=0001 1111, di conseguenza i 5 bit di configurazione meno significativi definiscono con il loro valore 1 un ruolo da ingresso per i pin di posizione meno significativa cioè RA0, RA1, RA2, RA3 e RA4. Analogamente, i bit più significativi (che occupano le posizioni 5, 6 e 7) individuano il comportamento da uscita per RA5, RA6 e RA7. L'esempio consente di chiarire qualunque residuo dubbio circa la corrispondenza tra bit del registro di direzione (TRISx), bit della porta X di I/O e pin del controllore.

Figura 24: lampeggio Led ver.g. Per accedere direttamente ad un solo bit di una porta è anche possibile utilizzare la notazione PORTx.By al posto della notazione PORTx.y.

Listato 5: lampeggio Led–ver.e

Il programma, il cui listato è mostrato in **figura 22** è analogo alle precedenti versioni a, b, c e d con l'unica differenza di utilizzare la notazione binaria per l'assegnazione del valore ai bit delle porte di uscita. Il prefisso % indica che la sequenza che segue è di tipo binario. Per esempio l'istruzione
PORTB:= %00110011;
comporta l'assegnazione RB0=RB1=1, RB2=RB3=0, RB4=RB5=1, RB6=RB7=0 essendo ad RB0 assegnato il bit meno significativo (cioè all'estrema destra) e a RB7 quello più significativo (cioè all'estrema sinistra).

Listato 6: lampeggio Led–ver.f

Il programma, il cui listato è mostrato in **figura 23**, è leggermente differente dalle precedenti ver. a, b, c, d ed e, e ha come scopo quello di mostrare come sia possibile eseguire un accesso diretto al singolo bit di un registro di direzione di una porta o della porta stessa. Per esempio:
TRISA.1:= 0;
significa (bit 1 di TRISA)=0 ==>imposta A1 come output.
Analogamente:
PORTA.1:= 0;
significa (bit 1 della porta A)=0 ==> pin A1=0.

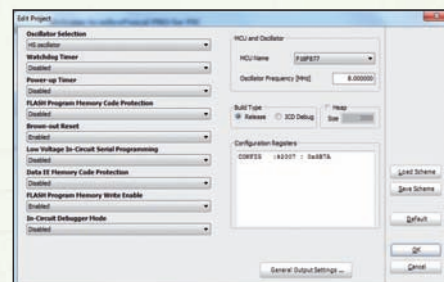



Figura 25: form di configurazione Edit Project.

Listato 7: lampeggio Led–ver.g

Il programma **(figura 24)**, come la precedente ver.f, mostra come sia possibile eseguire un accesso diretto al singolo bit di un registro di direzione di una porta o della porta stessa. Al posto della notazione
PORTx.y
usa però la notazione
PORTx.By
dove x individua la porta e può qundi essere A,B,C,D opp. E, mentre y indica il bit della porta. Per esempio, al posto di PORTA.1 potremo scrivere PORTA.B1. Stessa cosa vale per i registri TRISx. Per esempio, al posto di TRISA.1 potremo scrivere TRISA.B1. Sono fondamentalmente due modi diversi di scrivere la stessa cosa.

NOTAZIONE DECIMALE, ESADECIMALE E BINARIA

Abbiamo visto come un valore intero possa essere espresso in formato decimale, esadecimale o binario. Riassumiamo brevemente il concetto:
1. se non si usa alcun prefisso si intende che si sta esprimendo un valore in notazione decimale;
2. il prefisso $ o 0x indica che quello che segue è un numero espresso in notazio-

PER approfondire...

- [1] PIC16F87X Data Sheet 28/40-Pin 8-Bit CMOS FLASH Microcontrollers (www.mikrochip.com);
- [2] Sito di Microchip (www.mikrochip.com);
- [3] MikroPascal PRO for PIC v.5.61 demo e documentazione relativa (www.mikroe.com);
- [4] Help in linea MikroPascal Pro for PIC v. 5.61.

ne esadecimale per cui, per esempio, $8F oppure 0x8F rappresentano lo stesso valore;
3. il prefisso % indica che quello che segue è un numero espresso in notazione binaria. Ne consegue che tra 11, $11, 0x11, %11 solo lo seconda e terza espressione rappresentano lo stesso valore (pari a 17) essendo entrambe notazioni esadecimali; al contrario, la prima e la quarta rappresentano valori differenti (pari rispettivamente a 11 e a 3).

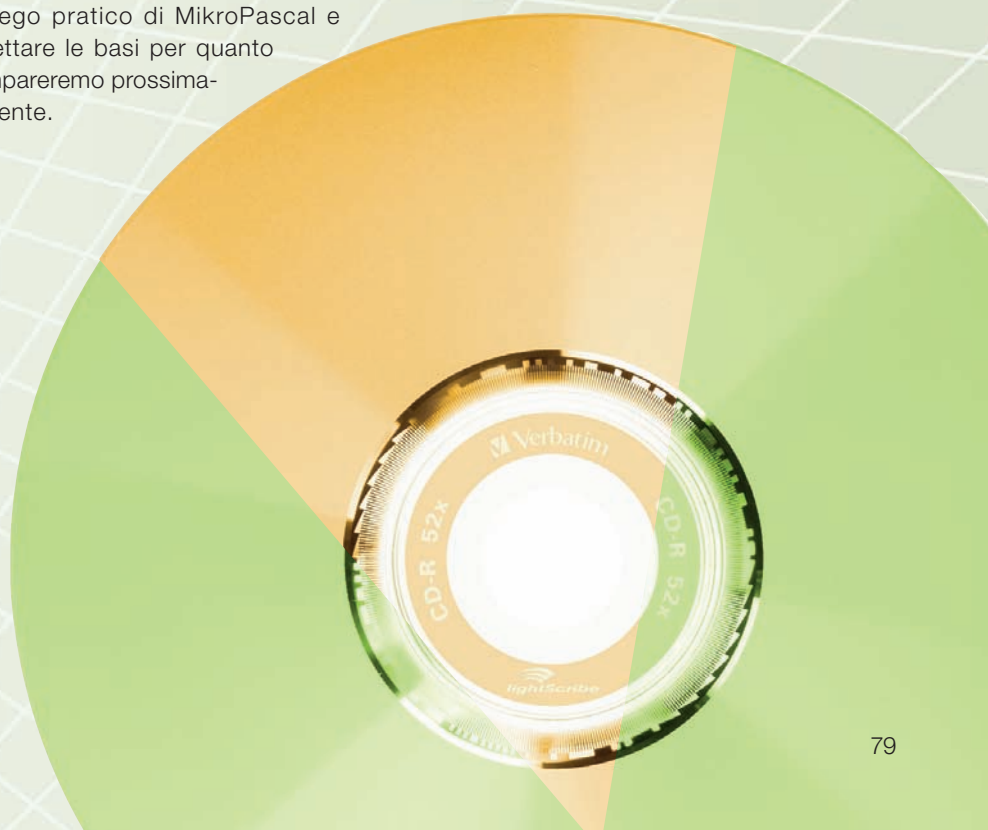
OSSERVAZIONI E CONCLUSIONI

Un progetto MikroPascal è sempre organizzato in un file con estensione ".mpppi". Il file di progetto tiene traccia del Target Device cioè il modello di controllore per il quale viene compilato il codice sorgente, dell'impostazione della frequenza di clock e della lista dei file sorgenti che compongono il progetto.
Trai i file del progetto abbiamo in particolare il file ".mpas", acronimo di MikroPascal, che altro non è se non il listato del programma contenuto tra le parole chiavi begin ed end.
Compilazione, linking ed ottimizzazione avvengono automaticamente con il comando Build [Ctrl+F9]. Se si desidera cambiare il modello di controllore per il quale eseguire la compilazione oppure la frequenza di clock prima della compilazione stessa è sufficiente aprire la finestra Edit Project (**figura 25**, Project->Edit Project [Ctrl+Shift+E]) e agire opportunamente sulle voci di configurazione del menu "option". Per modificare il modello di microcontrollore è sufficiente agire attraverso il menu a tendina MCU Name ***(figura 25)***, per modificare la frequenza di clock è invece sufficiente agire sul box Oscillator Frequency. È inoltre possibile impostare o meno il debugger. Quanto visto sulla notazione puntata per l'accesso al singolo bit di una porta o di un registro è un aspetto del tutto generale in MikroPascal. Con la stessa notazione è possibile accedere ai singoli bit di una variabile.
È più che mai opportuno osservare come negli esempi proposti non abbiamo utilizzato alcun ingresso ma sole uscite. Un sistema di controllo di un processo che consenta di eseguire delle operazioni senza decidere e percorrere il flusso di processo in base alla "percezione" e quindi l'analisi di segnali provenienti dall'esterno è come una macchina con i "paraocchi". Nella prossima puntata vedremo come acquisire degli stati logici in ingresso sulla base dei quali decidere istante per istante il flusso di processo facendo divenire il nostro controllore a tutti gli effetti un dispositivo intelligente di I/O e quindi di controllo.
In conclusione, in questa seconda puntata dedicata alla programmazione dei controllori PIC in MikroPascal, abbiamo imparato come è strutturato nelle linee generali un codice Pascal e come pilotare con livelli logici 1-0 i pin delle porte del controllore (qualora vengano configurate come uscite digitali). Abbiamo anche imparato a utilizzare la procedura guidata per la creazione di un nuovo progetto.
Abbiamo quindi fornito una serie di indicazioni su come è strutturato un codice Pascal e fatto una serie di osservazioni sulle segnalazioni di errore più elementari che il compilatore può generare in fase di compilazione. Abbiamo inoltre cominciato a prendere confidenza con i registri di direzione delle porte e con la rappresentazione esadecimale e binaria dei dati da scrivere in questi registri e sulle porte di I/O. Quanto mostrato non è di particolare complessità tuttavia ci ha dato la possibilità di rompere il ghiaccio con l'impiego pratico di MikroPascal e gettare le basi per quanto impareremo prossimamente.

NELLE PROSSIME PUNTATE

Nelle prossime puntate del corso mostreremo come sia possibile testare i codici di programma mediante l'impiego della famosa EasyPIC, daremo alcuni cenni sulla possibilità di testare il codice, anche senza disporre di alcuna board, mediante strumenti di simulazione, come fisicamente caricare il firmware sulla memoria flash del controllore utilizzando l'applicativo mikroProg Suite For PIC e la EasyPIC oppure altre schede o programmatori di terze parti. Continueremo quindi il percorso pratico sull'impiego dei controllori PIC e sulla loro programmazione mediante MikroPascal utilizzando i pin del controllore sia come ingressi sia come uscite. Riusciremo in questo modo a utilizzare il controllore come strumento in grado di elaborare stati logici acquisiti in ingresso, al fine di pilotare altri circuiti in uscita secondo uno specifico algoritmo. Indagheremo quindi alcuni aspetti hardware che riguardano rispettivamente il pin MCLR, il clock e la programmazione in-circuit e proseguiremo oltre, mostrando applicazioni via via sempre più interessanti. ❏

CODICE MIP 2828852

Tutorial PIC18 *(parte quinta)*

Da DIGITALE ad ANALOGICO, O QUASI

È noto che i microprocessori comprendono solo le cifre 1 e 0, ma spesso si ha la necessità di attivare dispositivi che invece prediligono una tensione variabile.

Nella lezione precedente abbiamo visto come un segnale analogico può essere convertito in forma numerica, quindi digitale, per essere elaborato tramite un microcontrollore. Non sempre però questi hanno convertitori DA, che sono in grado di trasformare l'informazione digitale in un segnale analogico, quindi nel caso si presenti la necessità di rendere analogica un'informazione digitale si rende necessario l'utilizzo di componenti esterni appositi; oppure se non si hanno grandi necessità, si può pensare di utilizzare la periferica del PIC che permette di generare segnali PWM.

IL CIRCUITO

Il circuito è lo stesso della lezione precedente, al quale vanno aggiunti solamente un diodo LED e una resistenza, da collegare al pin RC2 del PIC. Questo è il pin cui è collegata la periferica CCP1 che, tra le sue funzionalità prevede anche quella per la generazione di un segnale modulato in ampiezza, anche detto PWM.

CHE COS'È IL PWM

Il PWM è un tipo di modulazione digitale, acronimo di Pulse Width Modulazion, ovvero modulazione a larghezza d'impulso, che permette di ottenere una tensione media variabile al variare del Duty Cycle. Questo, solitamente espresso in percentuale, indica il rapporto tra la durata del semiperiodo a livello alto e la frequenza del segnale. Avremo quindi per un Duty Cycle al 50% un semi periodo a livello alto della durata di mezzo periodo del segnale PWM. Variando questa percentuale si può ottenere una tensione media variabile la quale può essere utilizzata per vari scopi come, ad esempio, pilotare un motore per variarne la velocità, controllare l'intensità luminosa di un LED o una lampadina oppure, aggiungendo all'uscita del generatore un filtro RC appositamente calcolato, generare un segnale analogico.

CONFIGURAZIONE DEL MODULO CCP

Per generare un segnale pwm, il PIC 18F2620 mette a disposizione i moduli CCP. Questo specifico modello ne ha due, i quali a loro volta utilizzano il Timer 2 per generare la frequenza del segnale. Per utilizzare questo timer si deve configurare il registro T2CON, in particolare impostando il Prescaler tramite i due bit meno significativi di questo registro, i cui valori possibili sono 1,4 e 16. Il terzo bit del registro T2CON, denominato TMR2ON, è utilizzato per attivare il Timer 2 semplicemente impostandolo a 1. Questo registro prevede anche altri 4 bit

TABELLA 1

T2CON							
-	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0
0	0	0	0	0	1	1	0

Tabella 1: registro T2CON e valori impostati.

di ROBERTO D'AMICO (IW0GTF)

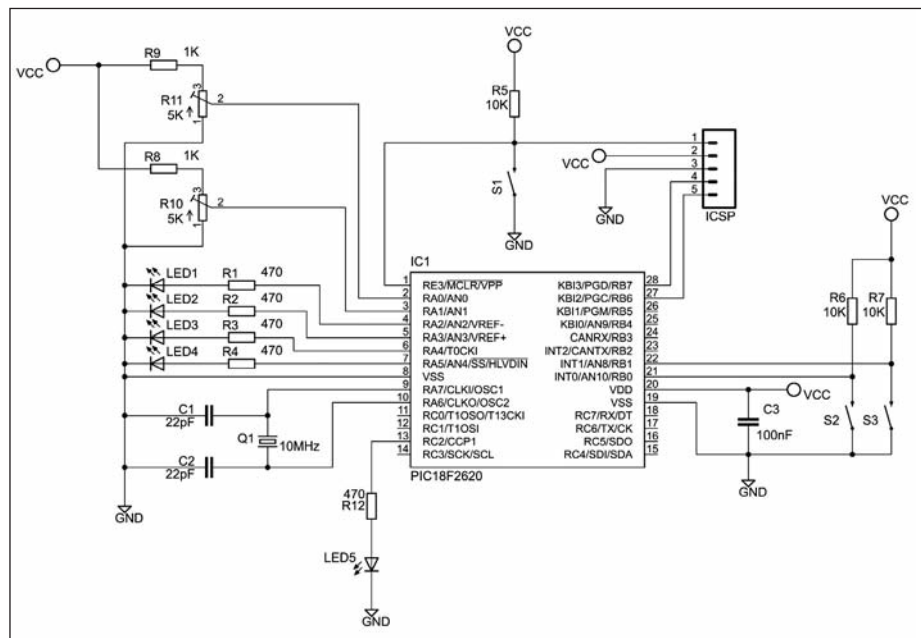



Figura 1: lo schema del circuito.

per la configurazione del Postscaler che non è utilizzato per la generazione della frequenza del PWM. Nella **tabella 1** si può vedere la composizione del registro e l'impostazione di alcuni valori per configurare il divisore del Prescaler a 16 e abilitare il Timer 2 al funzionamento.
Per determinare la frequenza di lavoro del Timer 2, oltre a impostare il valore del Prescaler, va definito anche il valore del registro PR2. Nel data-sheet è riportata una formula per calcolarne il periodo in funzione dei valori che si intende utilizzare nei registri.

```
PWMPeriod=
(PR2+1)*4*Tosc*TMR2PrescalerValue
```

Va tenuto in considerazione che esiste una frequenza minima al disotto della quale non è possibile scendere. Questo può essere facilmente verificato applicando alla formula i valori massimi possibili per il registro PR2 e per il Prescaler. Sostituendo i valori nella formula otteniamo:

```
PWMPeriod=
(255+1)*4*0,000000025*16=0,0004096
```

che corrisponde a una frequenza di 2,44KHz ed è la frequenza minima che possiamo ottenere. Se si avesse la necessità di avere una frequenza più bassa, l'unico punto su cui si può intervenire è agendo sulla frequenza del PIC per ridurla: è sufficiente dimezzarla per dimezzare anche la frequenza generata dal Timer 2.
Dopo aver definito la frequenza di funzionamento del timer, va configurato il registro CCPxCON per impostare il funzionamento della periferica per la generazione del PWM, perché questa periferica può funzionare anche in altre modalità. Per selezionarla sono utilizzati gli ultimi quattro bit di questo registro che, nel caso del PWM vanno valorizzati con 11xx. Gli ultimi due bit in questo caso possono assumere qualsiasi valore, come riportato nella **tabella 2**. DCxB1 e DCxB0 rappresentano i primi due bit dei 10 utilizzati per definire il valore del Duty Cycle.
Il registro CCPRxL rappresenta la parte alta dei 10 bit del registro utilizzato per impostare il periodo del PWM.

TABELLA 2

CCPXCON							
-	-	DCXB1	DCXB0	CCPXM3	CCPXM2	CCPXM1	CCPXM0
0	0	0	0	1	1	0	0

Tabella 2: registro CCPxCON e valori impostati.

VARIAMO LA LUMINOSITÀ DI UN LED

Nel **listato 1** è riportato il programma di esempio che utilizza un segnale PWM per variare la luminosità del diodo LED collegato al pin RC2. Come abbiamo detto in precedenza, variando il Duty Cycle aumenta la tensione media applicata al LED. Per questo scopo si può utilizzare la frequenza più bassa che si può generare, dato che è comunque molto più alta della frequenza che l'occhio umano riesce a distinguere. Quindi non avremo modo di vedere il lampeggio del LED, anche se ciò avviene effettivamente, come si può osservare nella **figura 3** dove, tramite un oscilloscopio, è visualizzato il segnale generato dal PIC.
Per variare l'intensità luminosa del LED useremo il valore di tensione di uno dei trimmer, letto tramite il convertitore AD, come già visto nella puntata precedente. Per definire il Duty Cycle del PWM, variando la posizione del cursore del trimmer avremo una variazione di luminosità del LED.
Poiché è utilizzato il convertitore AD, gran parte del codice del **listato 1** è simile a quello pubblicato nella puntata precedente, al quale è stata aggiunta tutta la parte di configurazione del Timer 2 e del registro CCP1CON. In particolare troviamo l'impostazione del registro PR2 al valore 0xFF che, insieme alla configurazione del bit T2KCPS1, serve per definire la frequenza di funzionamento del Timer 2 che è abilitato settando il bit TMR2ON.
Oltre al registro CCP1CON usato per definire la modalità di funzionamento, è inizializzato anche l'oggetto CCPR1, che rappresenta i 10 bit utilizzati per definire il Duty Cycle. Ultimo punto, ma non meno importante è la definizione della porta

LISTATO 1

```
#include <p18f2620.h>
#include <delays.h>
/**********
* DEFINE *
**********/
#define LED1 LATAbits.LATA2
#define LED2 LATAbits.LATA3
#define LED3 LATAbits.LATA4
#define LED4 LATAbits.LATA5
#define PULSANTE1 PORTBbits.RB0
#define PULSANTE2 PORTBbits.RB1
/********
* MAIN *
********/
void main(void)
{
   unsigned int result;
   /* Impostazione dei registri di configu-
razione del PIC per abilitare ADC sui pin
RA0 e RA1 */
   ADCON1=0b00001101;
   ADCON2=0b00000010;/*Left align*/
   ADCON0=0b00000001;
   CMCON=0b00000111;
   // Configurazione registri per PWM su
CCP1
   PR2=0xFF;
   CCPR1=0; /*Rappresenta i 10bit del reg-
istro per il Duty Cycle*/
   T2CONbits.T2CKPS1=1; /*Prescaler a 16*/
   T2CONbits.TMR2ON=1;
   CCP1CON=0b00001100;
   // Reset iniziale delle porte
   PORTA=0;
   PORTB=0;
   PORTC=0;
   // Definizione della funzione delle sin-
gole porte
   TRISA=0x03;
   /* I primi due bit della PORTB sono con-
figurati
      come ingressi a cui sono collegati i
pulsanti */
   TRISB=0b00000011;
   TRISC=0x00;
   // Accende il LED4 per indicare che il
programma è avviato
   LED1=1;
   LED2=1;
   LED3=1;
   LED4=1;
   Delay10KTCYx(250);
   LED1=0;
   LED2=0;
   LED3=0;
   LED4=0;
   // Loop infinito
   for(;;)
   {
       ADCON0bits.GO=1;
       while(ADCON0bits.GO);
       CCPR1L=ADRESH;
       CCP1CONbits.DC1B1=(ADRESL>>7) &
0b00000001;
       CCP1CONbits.DC1B0=(ADRESL>>6) &
0b00000001;
   }
}
```

RC2 come output con l'istruzione TRISC=0.
Rispetto alla lezione precedente, per quanto riguarda la configurazione del convertitore c'è una piccola differenza relativa all'allineamento del risultato della conversione, che in questo esempio è a sinistra. Questa variazione è usata allo scopo di semplificare il codice, in questo modo possiamo associare al registro CCPR1L, che rappresenta la parte alta del valore che definisce il Duty Cycle, direttamente il registro ADRESH, il quale a sua volta rappresenta la parte alta del risultato della conversione. Gli altri due bit, che figurano come i due più significati del secondo registro riportante il risultato del convertitore AD, ovvero ADRESL, sono estratti effettuando uno shift a destra, in funzione della posizione del bit e mettendoli in AND logico con il valore 0x01 per azzerare gli altri bit e utilizzarli per valorizzare i bit DC1B1 e DC1B0, che sono i due bit meno significativi per la definizione del Duty Cycle.

Figura 2: il circuito montato sulla bread-board.

LE FUNZIONI DEL C18 PER LA GESTIONE DEL PWM

Anche per i moduli CCP e la generazione di segnali PWM, le librerie del C18 mettono a disposizione delle funzioni per la loro gestione. Il **listato 2** propone uno spe-

LISTATO 2

```
#include <p18f2620.h>
#include <delays.h>
#include <pwm.h>
#include <timers.h>
/**********
 * DEFINE *
 **********/
#define LED1 LATAbits.LATA2
#define LED2 LATAbits.LATA3
#define LED3 LATAbits.LATA4
#define LED4 LATAbits.LATA5
#define PULSANTE1 PORTBbits.RB0
#define PULSANTE2 PORTBbits.RB1
/********
 * MAIN *
 ********/
void main(void)
{
    unsigned int pwmdc=500;
    /* Impostazione dei registri di configu-
razione del PIC per abilitare ADC
       sui pin RA0 e RA1 */
    ADCON1=0b00001101;

ADCON2=0b00000010;/*Left*///0b10000010;/*Right
*/
    ADCON0=0b00000001;
    CMCON=0b00000111;
    // Reset iniziale delle porte
    PORTA=0;
    PORTB=0;
    PORTC=0;
    // Definizione della funzione delle sin-
gole porte
    TRISA=0x03;
    /* I primi due bit della PORTB sono con-
figurati
       come ingressi a cui sono collegati i
pulsanti */
    TRISB=0b00000011;
    TRISC=0x00;
     // Accende il LED4 per indicare che il
programma è avviato
    LED1=1;
    LED2=1;
    LED3=1;
    LED4=1;
    Delay10KTCYx(250);
    LED1=0;
    LED2=0;
    LED3=0;
    LED4=0;
    // Configurazione Timer2
    OpenTimer2(TIMER_INT_OFF & T2_PS_1_16);
     // Attivazione PWM ed impostazione fre-
quenza
    OpenPWM1(0xFF);
    // Loop infinito
    for(;;)
    {
        // Impostazione Duty Cycle
        SetDCPWM1(pwmdc);
        if(PULSANTE1==0)
        {
            if(pwmdc <1020)
            {
                pwmdc = pwmdc +10;
            }
            while(PULSANTE1==0);
        }
        if(PULSANTE2==0)
        {
            if(pwmdc >10)
            {
                pwmdc = pwmdc -10;
            }
            while(PULSANTE2==0);
        }
    }
}
```

cifico programma per capire come vanno utilizzate. Rispetto al **listato 1** in questo programma troviamo le include alle librerie pwm.h e timers.h. Come abbiamo già visto nelle puntate precedenti, le librerie contengono istruzioni e definizioni necessarie per utilizzare delle specifiche periferiche. In questo esempio, le due periferiche necessarie per il funzionamento del PWM, sono inizializzate tramite le funzioni OpenTimer2 e OpenPWM1. La funzione OpenTimer2, com'è facile intuire, è utilizzata per configurare il Timer2 e richiede come parametro la combinazione in AND delle label, definite nel file timers.h, che rappresentano le possibili combinazioni del registro T2CON. Nel caso del **listato 2** sono utilizzate solo le due opzioni per disabilitare gli interrupt del Timer2 e per impostare il valore del prescaler. La funzione OpenPWM1, definita a sua volta nel file pwm.h, richiede come parametro lo stesso valore che nel **listato 1** abbiamo utilizzato per impostare il registro PR2, per definire la frequenza del segnale PWM.
Con le due istruzioni che abbiamo appena visto, la periferica per la generazione del PWM è pronta. A questo punto non rimane da definire che il valore del Duty Cycle. Per farlo si utilizza la funzione SetDCPWM1, che troviamo come prima istruzione del loop infinito e alla quale è passato come parametro il valore della variabile pwmdc, definita all'inizio del programma con un valore iniziale di 500 il quale rappresenta circa la metà del valore massimo supportato.
Il resto del codice, inserito all'interno del ciclo for, serve per controllare il valore della variabile pwmdc tramite la pressione dei pulsanti. Con il PULSANTE1 si incrementa il valore della variabile di controllo, mentre con il PULSANTE2 si decrementa. Analizzando il codice vedia-

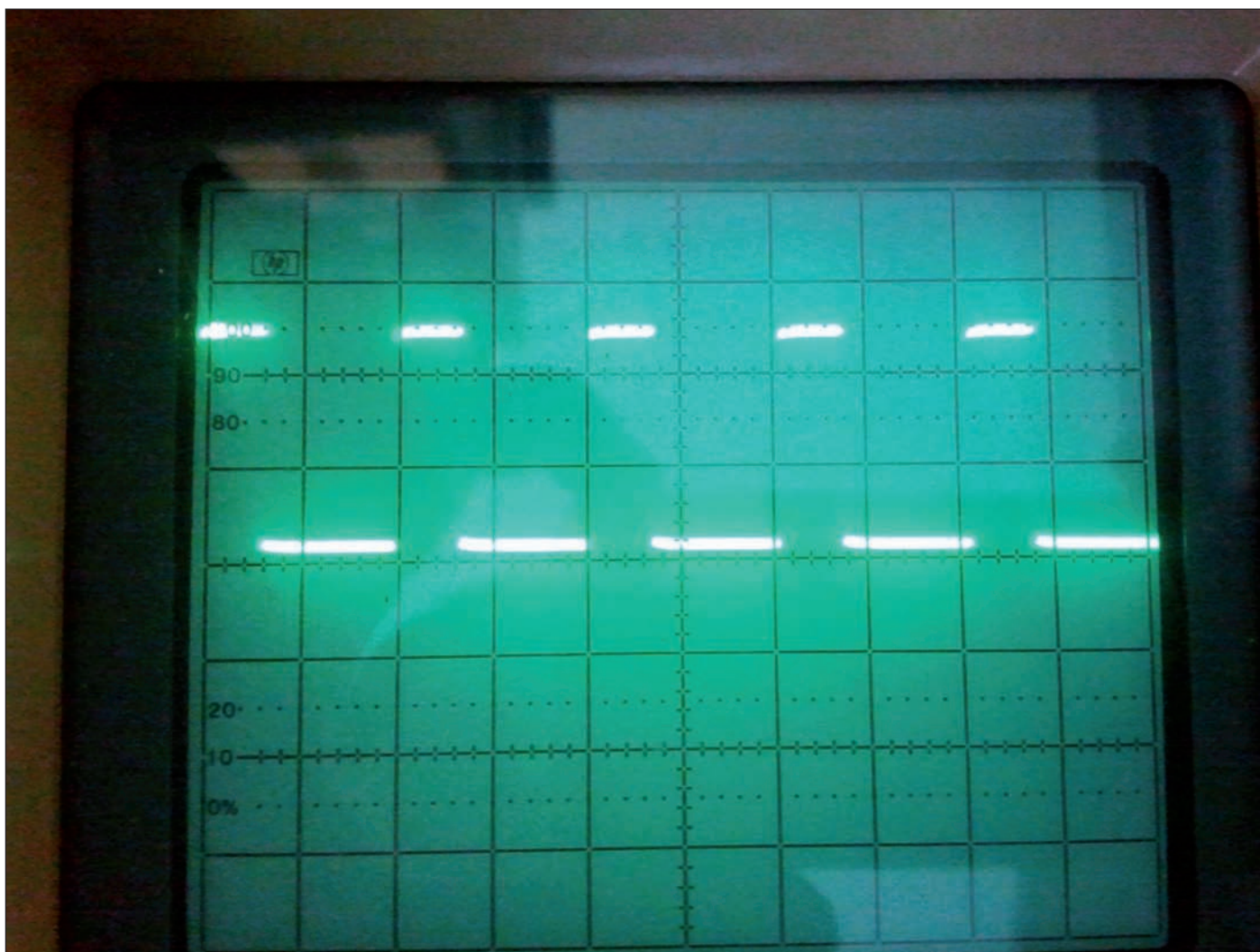



Figura 3: il segnale PWM generato dal PIC.

mo che il primo if controlla lo stato del PULSANTE1. Quando questo è premuto, quindi l'ingresso del PIC associato a PULSANTE1 è uguale a 0, l'esecuzione del programma passa all'interno del blocco di codice contenuto nelle parentesi graffe ed esegue il controllo if successivo, il quale serve a impedire che la variabile pwmdc possa essere incrementata oltre il valore massimo. Teoricamente questo valore, essendo un numero a 10 bit, corrisponde a 1023, nell'esempio è usato 1020 per semplificare, visto che gli incrementi sono effettuati a multipli di 10. Il controllo while che segue serve per attendere che il pulsante sia rilasciato. Come già visto nelle altre puntate, questo controllo va bloccare l'esecuzione del programma all'interno del suo blocco istruzioni, che nel nostro esempio comprende la sola istruzione while. Questa forzatura al codice, introdotta dal controllo while, permette di evitare che la variabile sia incrementata fino al suo valore massimo alla pressione del pulsante, perché come abbiamo già visto in altri casi, il codice è eseguito molto più velocemente di quanto il dito possa rilasciare il pulsante.
La sezione successiva del codice esegue la stessa tipologia di controlli per decrementare il Duty Cycle, tramite la pressione del PULSANTE2.

CONCLUSIONI

Abbiamo visto in questi due esempi di codice come si può generare un segnale PWM con un PIC e come variarne il Duty Cycle. Il PIC 18F2620 dispone di due periferiche simili. Come si può notare nel **listato 2**, anche le istruzioni OpenPWM e SetDCPWM hanno un numero che le identifica e, per utilizzare il secondo canale PWM, non occorre fare altro che richiamare le stesse istruzioni con il numero 2 alla fine. Va ricordato però che entrambe le periferiche usano il Timer2, quindi dovranno per forza di cose avere la stessa frequenza, ma potranno avere Duty Cycle differenti.
Se volete fare qualche esperimento, avrete notato che eseguendo il codice del **listato 2**, il LED non è mai definitivamente spento. Provate a modificare il codice agendo solo sulla regolazione del Duty Cycle per fare in modo che ci sia la condizione per cui con la pressione del PULSANTE2 si arrivi a spegnerlo definitivamente. ❏

CODICE MIP 2821443

Interfacciamento dei processori *(parte undicesima)*

GESTIONE CONTEGGIO *su display a due digit*

La gestione diretta dell'informazione nella nativa forma esadecimale, su display a due o più digit, è operazione che i microcontrollori e i dispositivi programmabili del PC eseguono con una certa complessità: l'interfaccia con 9368 rende facile e immediato questo compito.

Nelle puntate precedenti abbiamo sottolineato che la gestione dei visualizzatori, direttamente collegati sulla porta d'uscita di un qualunque dispositivo programmabile (microcontrollore o personal computer), risulta facilitata aggiungendo uno strato di interfaccia costituito da due decoder, in grado di servire due digit alla volta con un solo byte, alimentando con i suoi due nibble le rispettive quaterne di linee d'ingresso dei decoder.

Affidando il compito ai TTL 74LS47 e 74LS48 o al CMOS 4511 abbiamo scoperto che essi sono in grado di interpretare solo le prime 10 delle 16 combinazioni possibili di ciascuna quaterna di bit (cioè sono adatti a decodificare solo il Codice BCD, come noto costituito da 10 elementi, i numeri da 0 a 9 del sistema di numerazione decimale) mentre con TTL 9368 questo limite è superato, essendo l'unico componente in grado di interpretare un Codice Binario Puro a 4 bit e di mostrare tutti i 16 simboli del sistema di numerazione esadecimale.

La **figura 1a** mostra il suo pinout e ne evidenzia la struttura logica, decisamente

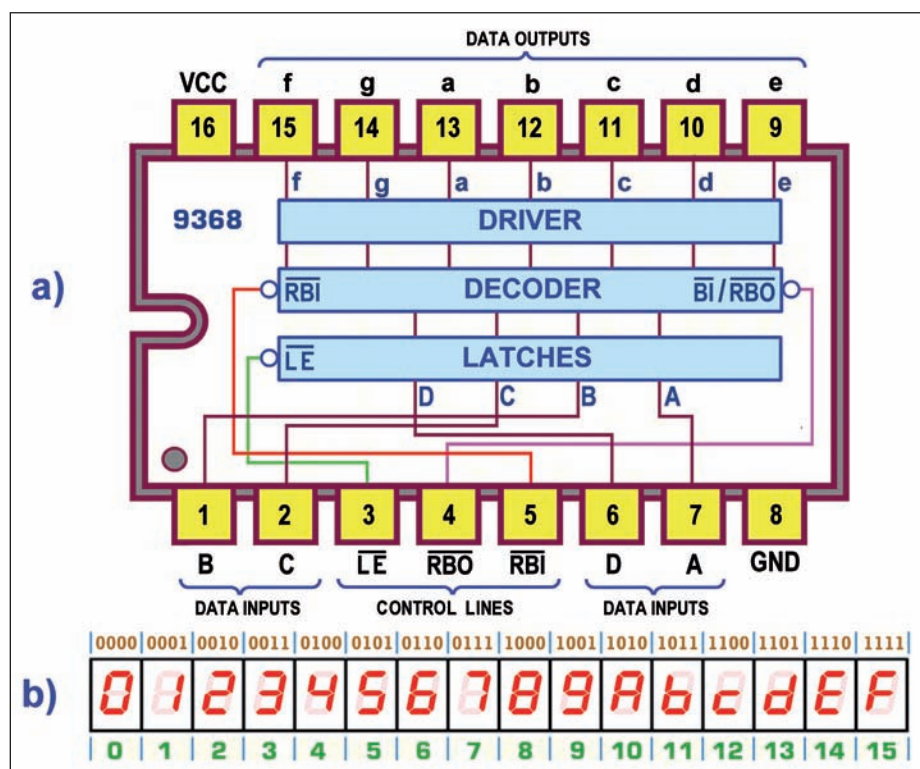



Figura 1: binary to 7-segment Latch/Decoder/Driver 9368: pinout e simboli visualizzati con FND500.

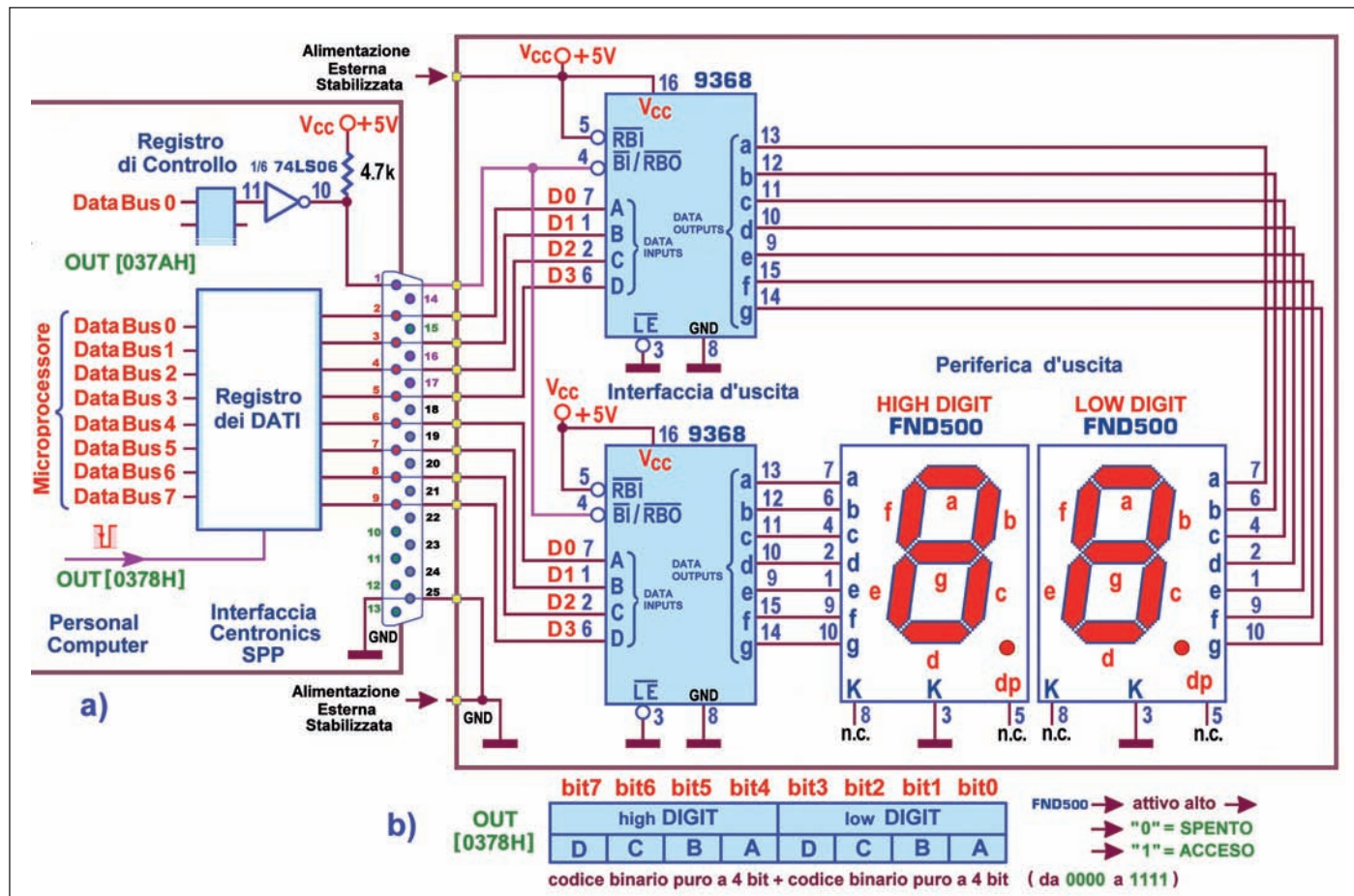



Figura 2: controllo di due FND500 con interfaccia d'uscita 9368, in uscita al Registro Dati SPP 0378H.

speciale: esso è in grado di dare (oltre al suo intrinseco servizio di Decoder adatto a pilotare i segmenti di un digit a catodo comune) anche la possibilità di memorizzare i 4 bit delle informazioni fornite sulle linee d'ingresso e la disponibilità di Driver su ciascuna delle 7 linee d'uscita attive alte, al fine di fornire le migliori condizioni di pilotaggio ai segmenti-LED ad esse collegati, cioè in grado di erogare verso di essi una corrente I_{OH} da 16 mA (minima) a 22 mA (massima), senza bisogno di resistori per limitare la corrente.
La memoria è costituita da 4 flip-flop di tipo D-Latch, controllati contemporaneamente sul livello della linea di sincronismo (LE**,** Latch Enable, pin3): quando LE è a massa (livello 0) i 4 bit del codice binario applicato in ingresso passano inalterati a valle della memoria, che risulta essere "trasparente" nei loro confronti; nell'istante in cui LE passa da un livello basso a uno alto (cioè sul suo fronte di salita) le uscite D-Latch "scattano" sulla quaterna di bit presente su di esse in quel momento, tenendola bloccata (memorizzandola) per tutto il tempo nel quale la linea LE è tenuta a livello 1, fino a quando verrà riportata a 0.
La **figura 1b** mostra i simboli del sistema di numerazione esadecimale offerti dal decoder per ciascuna delle possibili 16 combinazioni del Codice binario a 4 bit proposto in ingresso; su questa parte circuitale agisce l'influenza dei segnali BI**/**RBO (pin4) e RBI (pin5), entrambi attivi bassi, utilizzati per gestire più componenti 9368 in cascata, con la tecnica di Ripple Blanking, che evita di accendere tutti gli zero non significativi, qualora la cosa possa essere ritenuta necessaria, come nell'ambito di un visualizzatore decimale con numerose cifre.
Di questo abbiamo parlato ampiamente nell'ambito della scorsa puntata, nella quale abbiamo anche progettato un codice assembly in grado di controllare a basso livello l'intensità luminosa del visualizzatore (due digit FND500) con l'aiuto di un'onda quadra non simmetrica generata da software, affidando questa curiosa applicazione alle proprietà dell'ingresso BI, Blanking Input, tali da spegnere incondizionatamente il digit collegato al Decoder, con qualunque codice presente sugli ingressi DCBA.
Riprendiamo ora lo schema necessario per pilotare una coppia di digit FND500 (a catodo comune) con due 9368, affidato la volta scorsa alla porta PortA di un single-chip, riadattandolo per consentirne il controllo tramite il Registro Dati 0378H della porta parallela di un personal computer; ricordiamo che, nonostante sia sempre meno presente, essa è facilmente disponibile con l'aggiunta di un'economica

scheda di I/O e rimane pur sempre un eccellente strumento per interfacciare numerosi dispositivi esterni con la potenza indiscutibile di un PC, di gran lunga superiore a quella di un single-chip.
La **figura 2** ne mostra i dettagli insieme alla Tabella di Associazione Logica, che virtualizza la presenza dei due digit, offrendo ogni dettaglio utile alla creazione del programma necessario per la loro gestione senza dover analizzare lo schema elettronico a essi relativo: il suo contenuto propone la configurazione del byte da spedire sulla porta d'uscita, distinguendo in parte alta (High Digit) e parte bassa (Low Digit) il livello delle due quaterne da fornire agli ingressi dei due 9368.
Lo schema sottolinea che, nel normale funzionamento, il piedino di controllo LE (pin3) della memoria latch deve essere lasciato a massa mentre le due linee di controllo del Ripple Blanking, RBO (pin4) e RBI (pin5), irrilevanti in questo caso, possono essere lasciate inattive, lasciando scollegata la prima e scollegata (o meglio, collegandola a +5 V) la seconda. In virtù di una particolare circuiteria interna linea collegata al pin4 del 9368 può essere utilizzata, senza produrre danni, sia come uscita (rendendo disponibile il segnale RBO) sia come ingresso (consentendo di forzare su di essa il segnale BI): nello schema si nota che questa linea è affidata al bit0 del Registro di Controllo della parallela: sebbene non sia strettamente necessario, questo artificio permette (facoltativamente, con BI=0) lo spegnimento da hardware del digit collegato al Decoder.
In concreto, non essendo possibile disporre da software di questo servizio, questa rimane l'unica via, nel caso non se ne possa fare a meno.
Per collaudare questa struttura ho progettato un programma in grado di gestire in tutti i modi possibili un conteggio progressivo sulla periferica costituita dalla coppia di FND500 interfacciata dai due decoder/latch/driver 9368; il codice assembly 80x86 necessario per la sua gestione è offerto in **figura 3**.

Figura 3: codice assembly per la gestione di due Digit FND500 con interfaccia 9368.

Come si può notare si tratta di un programma molto semplice, ma sviluppato in modo decisamente attraente: dopo la consueta procedura (Desktop) che si occupa della presentazione a video dei compiti del progetto, si provvede alla necessaria inizializzazione delle variabili coinvolte, in questo caso la sola Depo, una locazione di memoria a 8 bit utilizzata dai vari servizi come punto di riferimento per prendere e inserire il byte da usare come informazione per i due nibble da fornire (via 9368) ai due digit; all'inizio questo byte è forzato a 00H così da accendere il visualizzatore con i simboli "00".
Segue la parte relativa al controllo dell'hardware (i due FND500 interfacciati con altrettanti 9368, che si appoggia alla procedura Out08bit, analizzata in dettaglio tra poco) mentre quella gestionale si sviluppa nell'ambito del Main Loop (cioè è affidata alle istruzioni comprese tra le due etichette xxx02) e organizza l'informazione proposta sul visualizzatore, di volta in volta diversa in funzione del tasto premuto sulla tastiera del PC, chiamato a mettere in atto il relativo Servizio; questa parte di codice è interessante perché insegna la tecnica migliore per interrogare e servire la pressione dei soli tasti autorizzati.
I Servizi previsti sono stati pensati per le tipiche necessità di una periferica come quella qui utilizzata e mettono in evidenza come da software si possano fare a costo zero quasi tutte le cose che farebbero circuiti digitali realizzati con componenti reali, di certo ingombranti e costosi; nel nostro progetto è prevista la disponibilità di:
a) un conteggio automatico in avanti (previa la pressione del tasto "U", Up) o indietro (con la pressione del tasto "D", Down);
b) un incremento manuale in avanti (con il tasto "I", Inc) o indietro (con il tasto "E", dEc);
c) l'azzeramento asincrono del valore corrente mostrato sui digit (con il tasto "R", Reset);
d) lo spegnimento di entrambe le cifre del visualizzatore (con il tasto "B", Blank), mantenendo comunque inalterato il valore corrente, dal quale si potrà ripartire con i servizi di conteggio.
La **figura 4** analizza il codice delle pro-

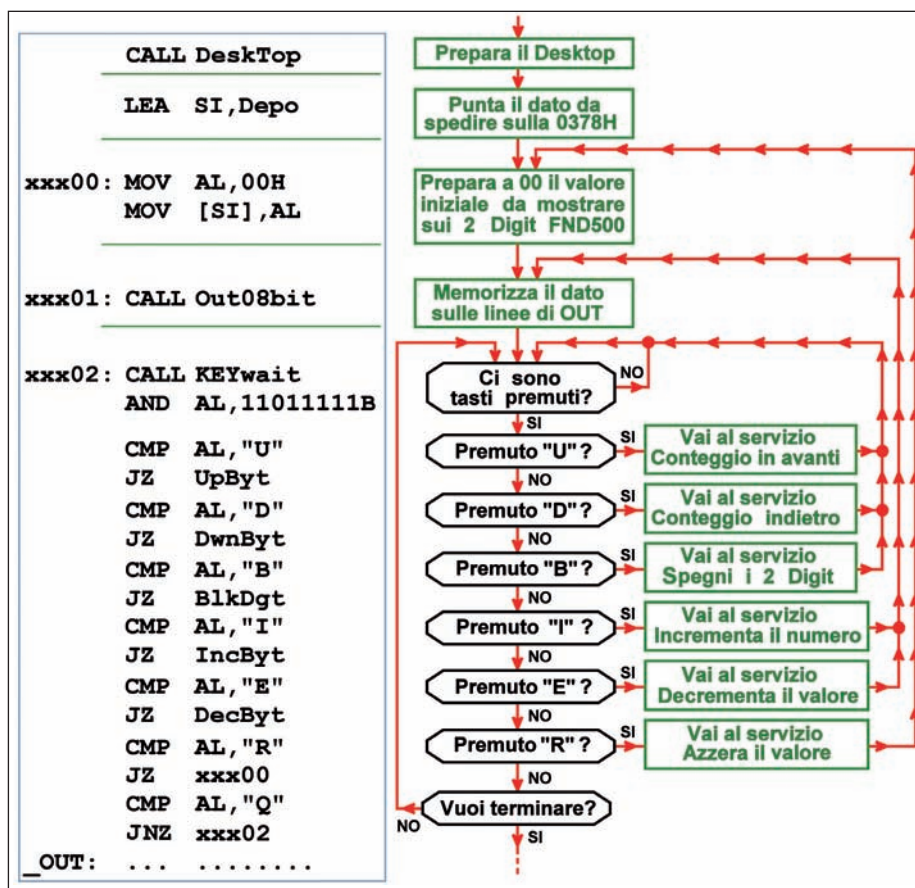
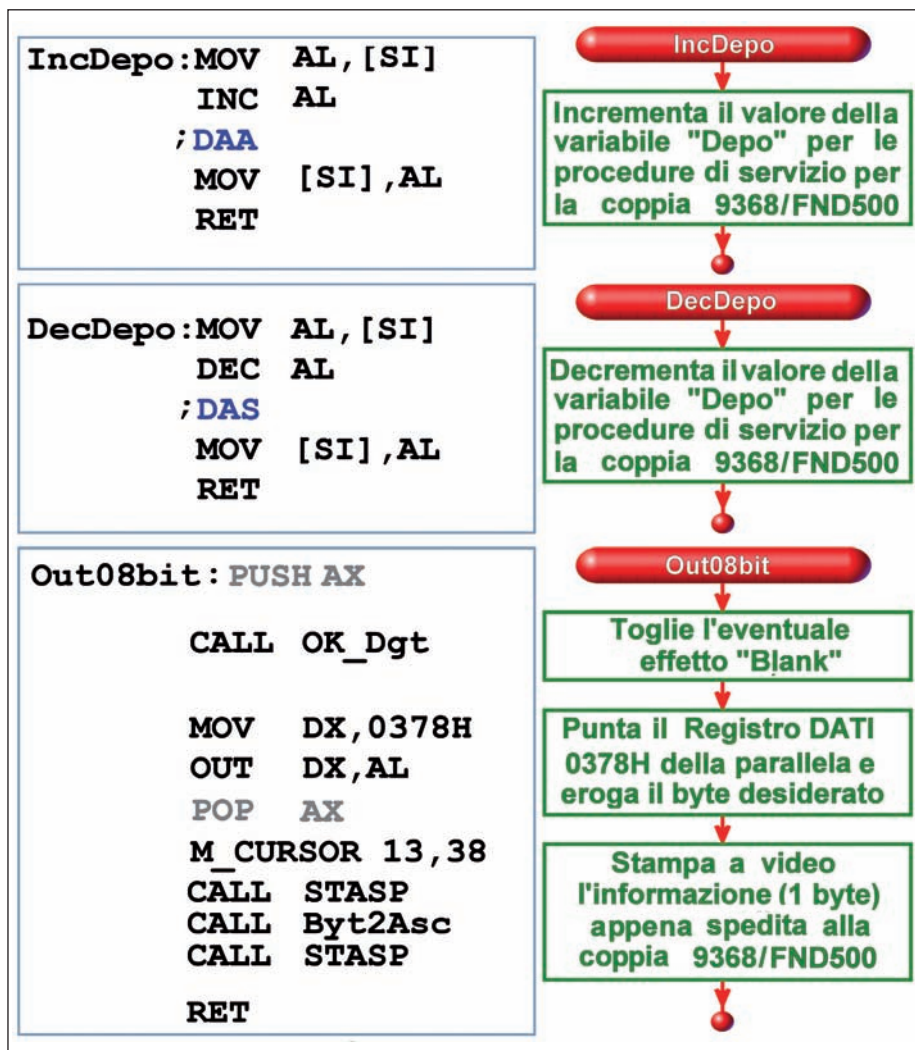



Figura 4: codice assembly per la gestione del conteggio sui due Digit FND500.

cedure principali; le prime due (IncDepo e DecDepo) sono predisposte per supportare i quattro servizi di conteggio (mostrati in **figura 6**); il loro compito si limita a prelevare il valore corrente dalla locazione Depo (puntata fin dall'inizio dal registro SI) e a provvedere rispettivamente al suo incremento o decremento di una unità, prima di rimetterlo al suo posto, a beneficio dei servizi citati.

Da notare la presenza delle istruzioni DAA e DAS: il ";" posto davanti a esse fa sì che l'assemblatore non le compili e che dunque il processore non le esegua; in questo modo il conteggio (in avanti o indietro) verrà condotto in esadecimale, sfruttando a pieno le capacità dei decoder 9368 così che, sui due Digit, alla visualizzazione .., "09" (associata al numero 09H) seguiranno le "0A", "0B", .. , "0F", "10", .., legate ai numeri 0AH, 0BH, .. ,0FH, 10H, ..

Togliendo il ";" posto davanti a esse e ricompilando il sorgente, le istruzioni DAA e DAS (due tra le più potenti del set del processore 80x86) saranno operative: il loro compito è quello di "aggiustare in decimale" il valore esadecimale dopo una somma o una sottrazione (nel nostro caso dopo un incremento o un decremento); in questo modo è possibile riadattare il codice macchina del nostro progetto per obbligarlo a mostrare sui due Digit un conteggio decimale (nonostante la natura esadecimale dei decoder 9368) così che la sequenza di coppie di simboli mostrata sarà del tipo: .., "09", "10", "11", .. , evitando di coinvolgere numeri che richiedono i sei simboli letterali del codice binario puro e provvedendo di fatto ad una visualizzazione BCD a due cifre.

La terza procedura (Out08bit) di **figura 4** è quella più importante, a livello hardware; dopo aver assicurato (CALL OK_Dgt) la presenza di un livello alto sulla linea **BI** dei due 9368 si limita a porre il valore corrente della variabile Depo (predisposto nel registro accumulatore AL dal codice chiamante) sul Registro Dati 0378H della porta parallela: immediatamente questo byte viene ripartito tra le due quaterne di linee d'ingresso dei due 9368 e, da essi, passato ai due digit, per l'istantanea visualizzazione.

La parte finale della procedura provvede a fare eco sul monitor di questo valore, in posizione centrale (riga 13, colonna 38) e con l'aiuto della potente procedura di conversione Byt2Asc, capace di tradurre un numero binario (non stampabile) in una stringa di caratteri ASCII stampabili.

La **figura 5** si occupa di un servizio collaterale ma non per questo banale; ho pensato di dare la possibilità di collaudare la linea d'ingresso BI (pin4 del 9368) intervenendo su di essa da software per forzarla a 1 (nel normale servizio di visualizzazione) o a 0 (per spegnere incondizionatamente i digit): la prima (la procedura OK_Dgt) è chiamata automaticamente dopo ogni richiesta di conteggio, mentre la seconda (il servizio BlkDgt) è imponibile solo su richiesta, premendo il tasto "B".

In entrambi i casi questa azione è resa possibile programmando il bit0 del Registro di Controllo 037AH della porta parallela; non bisogna dimenticare che il valore del bit è invertito internamente per cui, forzandolo a 0, la linea in uscita dal corrispondente pin1 del connettore della parallela passa alta e, in questo modo, forza a 1 l'ingresso BI di entrambi i 9368, producendo la riaccensione immediata di tutti i segmenti dei Digit; effetto opposto si avrà ovviamente forzando il bit0: linea bassa su pin1, ingresso BI=0, attivo, e spegnimento immediato di tutti i segmenti di entrambi i Digit; da notare che (per una attenta gestione dell'informazione) anche la stringa visualizzata sul monitor viene cancellata con un paio di spazi.

La **figura 6** descrive quattro dei sei Servizi previsti per la gestione di questo pro-

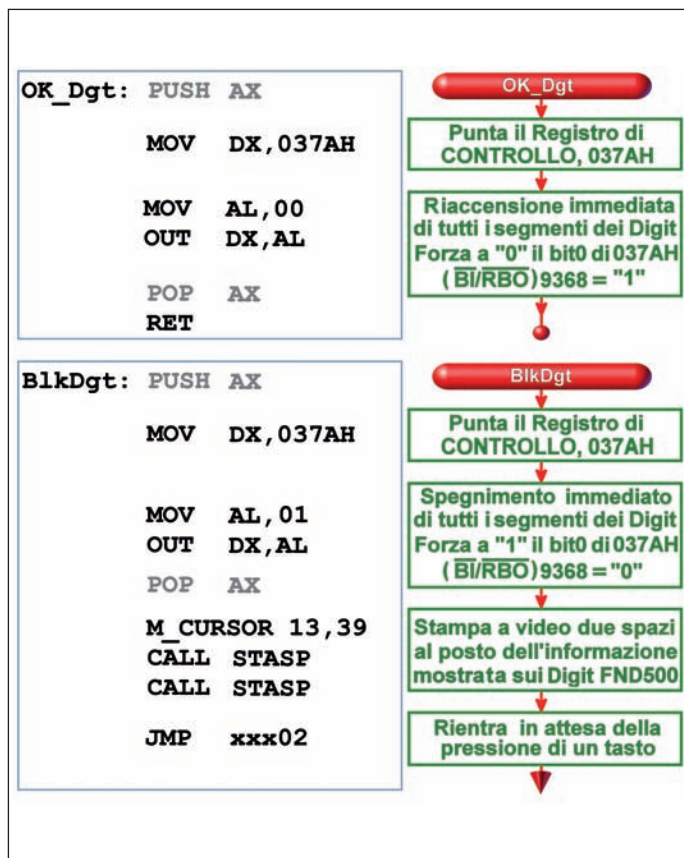



Figura 5: codice assembly per accendere e/o spegnere i due Digit FND500, con il controllo di BI.

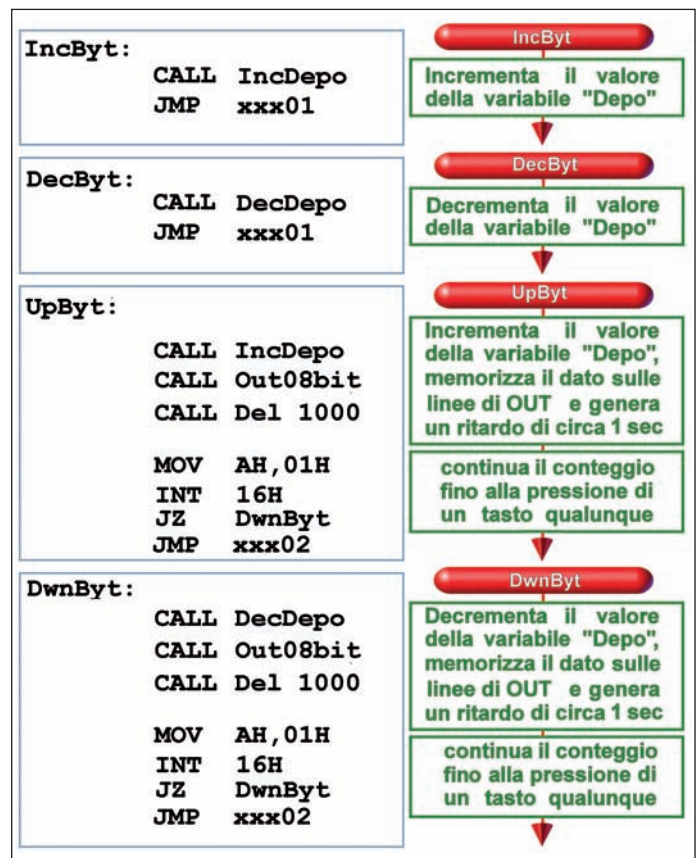



Figura 6: codice assembly per i quattro servizi dedicati alla gestione di un conteggio sui due Digit FND500.

getto: non si tratta di procedure ma di codice specificatamente pensato per rispondere propriamente alla pressione di un tasto autorizzato; a ciascuno di essi si accede infatti con un salto condizionato (dalla flag di Zero, JZ address, la cui presenza a 1 stabilisce che l'evento si è verificato) e vi si esce con un salto incondizionato (JMP address) in un punto antecedente il loop d'attesa (non esiste dunque RET).

La cosa è particolarmente evidente anche riguardando la **figura 3**; in dettaglio:

a) i primi due Servizi (IncByt e DecByt) si limitano semplicemente a incrementare o decrementare il valore mostrato sui digit (e sul monitor) su diretta richiesta della pressione dei tasti "I" e "E", rientrando poco prima (con un JMP a livello XXX01) del Main Loop per consentire la visualizzazione, prima di una nuova attesa di tasti premuti;

b) i rimanenti due Servizi (UpByt e DwnByt) sono più sofisticati: l'incremento o il decremento del valore mostrato deve continuare senza interruzioni, con cadenza di un secondo, per garantire un conteggio Up (attivato dalla pressione del tasto "U") o un conteggio Down (attivato dal tasto "D"); per questo è necessario appoggiarsi alla Funzione 01H della procedura a livello BIOS nota come INT 16H, che consente di verificare la eventuale pressione di tasti senza fermarsi; non appena viene rilevata la pressione di un tasto qualunque viene infatti imposto il rientro (con un JMP a livello XXX02) nel Main Loop.

Da notare che le temporizzazioni sono rese possibili dalla procedura Del_1000, in grado di generare un ritardo fisso di 1 secondo con particolare precisione, appoggiandosi per questo al Timer di sistema del personal computer.

Il Servizio dei due tasti rimanenti:

a) dopo la pressione del tasto "Q", Quit, si limita a uscire dal Main Loop, proseguendo via _Quit, terminando il programma di test e restituendo il controllo al sistema operativo;

b) dopo la pressione del tasto "R", Reset, provvede a reinizializzare il programma rientrando semplicemente a livello XXX00, dove la variabile Depo viene forzata a 00H così da garantire l'azzeramento asincrono del valore corrente mostrato sui digit.

La versione completa del codice Assembly (Deci9368.ASM, insieme a numerose note esplicative e all'eseguibile, pronto per l'uso, ottenuto dalla sua compilazione) è disponibile puntando il link di approfondimento consigliato. ❑

Data Acquisition System Design: SAMPLE & HOLD

Il sistema di acquisizione dati (DAQ) è un sistema di misura capace di captare, elaborare e registrare le misure di grandezze fisiche. In questo primo articolo di una serie analizzeremo i principali aspetti di un DAQ e le tecniche di progettazione, con l'biettivo di fornire non solo aspetti teorici avanzati ma anche nozioni pratiche utili ai progettisti di questo campo.

I sistemi di acquisizione dati necessitano di un convertitore Analogico/Digitale per convertire il segnale analogico (per esempio un sensore di temperatura, un sensore magnetico ecc.) nel mondo digitale. Per far ciò è necessario lasciare al convertitore ADC un certo tempo, in cui è essenziale che il segnale rimanga costante. Lo scopo del Sample&Hold (S&H) è proprio questo, cioè trasformare il segnale in ingresso in una serie di gradini di ampiezza pari al tempo di campionamento. L'operazione del S&H **(figura 1)** è suddivisa in due fasi che si alternano:

- una fase di aggancio, in cui il dispositivo insegue il segnale di ingresso (sample o track);
- una seconda fase in cui il segnale viene campionato e mantenuto costante (Hold).

Questa fase è eseguita mediante un condensatore. Il dispositivo ha un segnale di controllo che identifica la fase in cui il dispositivo si trova, un ingresso analogico (V1) e un'uscita (V2). Il segnale di controllo comanda un interruttore che quando è chiuso garantisce la fase di caricamento del condensatore, mentre quando è aperto "isola" il condensatore fino alla successiva fase di Sample.
A seconda della dimensione del condensatore possiamo avere due situazioni:

- se la capacità è elevata si ha una maggiore precisione, ma un maggior tempo di caricamento (minori prestazioni);
- se la capacità è di dimensioni limitate abbiamo poca precisione, ma migliori prestazioni. L'interruttore Q1, inoltre, è normalmente realizzato con tecnologia n-MOS, pilotato con una tensione che deve essere maggiore della tensione di soglia dell'interruttore stesso. Si deve scegliere una tensione di pilotaggio del MOS superiore a quella massima della dinamica per l'accensione e minore di quella minima per lo spegnimento.

Le imprecisioni di funzionamento del più semplice Sample&Hold sono legate agli stessi problemi che presentano i comparatori per via degli switch o interruttori (Q1). Lo switch, nell'aprirsi, causa un'i-

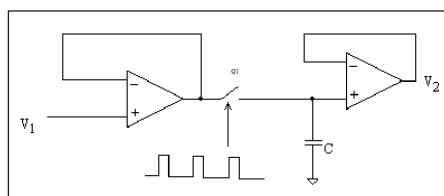



Figura 1: Sample&Hold.

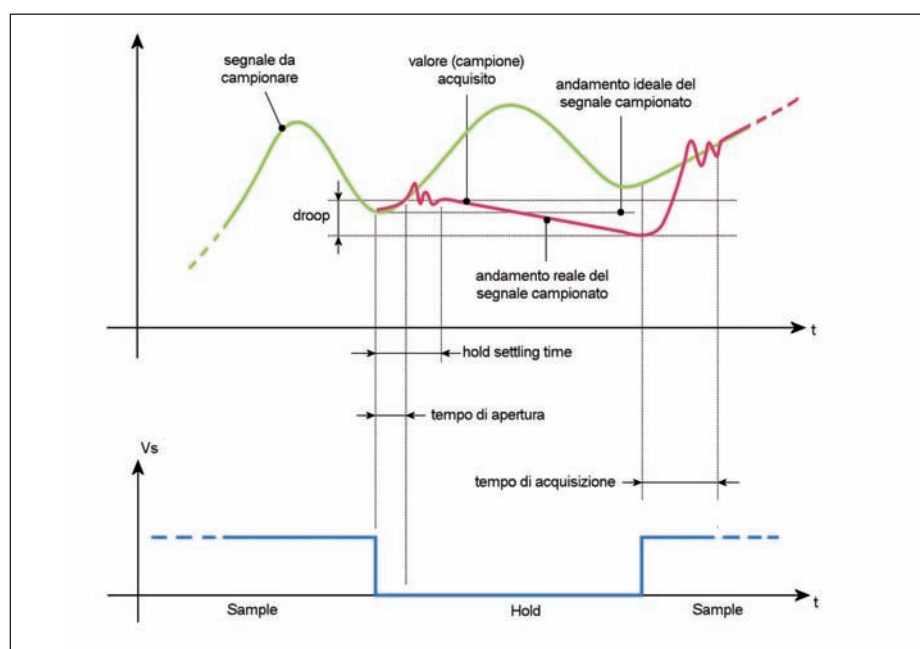



Figura 2: parametri del Sample/Hold.

di MAURIZIO DI PAOLO EMILIO

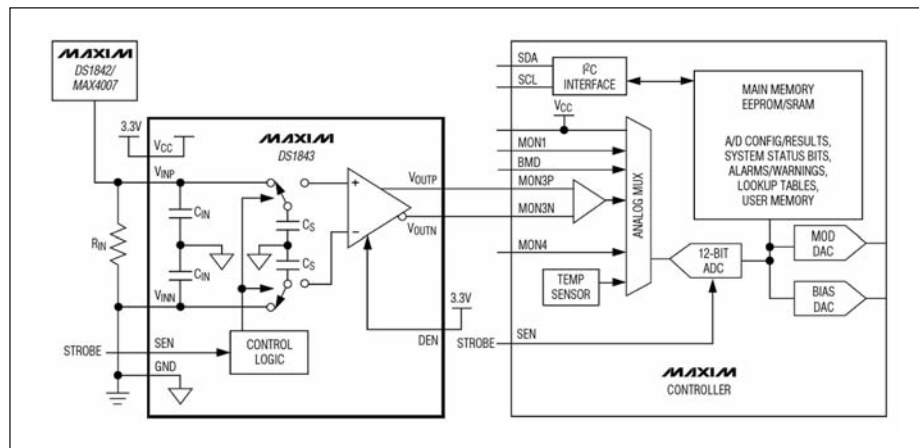



Figura 3: DS1843 di Maxim.

niezione di carica (in ingresso al secondo operazionale) modificando la tensione campionata. Le componenti di errore sono due, dovute al clock feedthrough e alla redistribuzione di carica.
Quest'ultimo caso è più grave, sia perché la carica è maggiore sia perché dipende dal segnale e introduce quindi una distorsione. Per ridurre gli effetti di iniezione di carica si può utilizzare un transfer gate, quindi sostituire lo switch con una coppia di transistor (Q1, pnome e nmos) pilotati da segnali opposti. In tal modo mentre il pmos immagazzina carica positiva e nmos carica negativa, i due effetti dovrebbero annullarsi.
In realtà non è quasi mai possibile riuscire ad annullare esattamente la carica del pmos e del nmos (che dipende dal segnale).
Un meccanismo che consente di migliorare questo effetto annullando la carica iniettata è l'utilizzo di un secondo switch (Q2, in serie all'altro) che è sempre cortocircuitato ma si accende quando l'altro si spegne.
In questo caso la cancellazione non è perfetta (soprattutto se il clock non ha fronti sufficientemente ripidi) ma è comunque più efficace (non dipende dal segnale). I due clock devono comunque essere leggermente sfasati per garantire che la carica del transfer gate Q2 non sfugga attraverso Q1.
Con riferimento alla **figura 2** i parametri più significativi di un Sample&Hold sono i seguenti:

Tempo di apertura t_{ap}

È l'intervallo di tempo che intercorre tra l'istante di attivazione del comando di Hold e l'istante in cui il condensatore è effettivamente scollegato dal segnale di ingresso. Il valore del campione effettivamente catturato corrisponde al valore dell'ingresso nell'istante finale dell'intervallo.

Aperture jitter t_j

Questo parametro può essere definito come l'intervallo di variabilità del tempo d'apertura e quindi è anch'esso un tempo. Sostanzialmente, il jitter è generato dal rumore della rete di commutazione e si ripercuote in variazioni del valore del campione effettivamente acquisito. È un parametro molto importante perché stabilisce la massima frequenza di campionamento consentita dal S/H utilizzato.

Tempo di assestamento della fase di hold (Hold Setting Time)

È l'intervallo di tempo che intercorre tra l'istante di attivazione del comando di hold e l'istante in cui l'uscita si stabilizza sul valore effettivamente campionato. L'ordine di grandezza di questo parametro è nell'intorno delle centinaia di nanosecondi.

Tasso di decadimento DR (Droop Rate)

Indica la variazione del valore del campione, effettivamente catturato e presente in uscita, provocata dalle correnti di perdita. L'ordine di grandezza di questo parametro varia da qualche unità alle decine di millivolt/secondo (mV/s).

Tempo di acquisizione t_{ac} (Acquisition Time)

È l'intervallo di tempo che intercorre tra l'istante di attivazione del comando di sample e l'istante in cui la tensione sul condensatore, cioè l'uscita, inizia effettivamente a seguire l'andamento della tensione di ingresso. L'ordine di grandezza di questo parametro varia da qualche microsecondo a qualche decina di microsecondi. Il Sample&Hold DS1843 di Maxim è un IC ideale per operazioni di questo tipo per segnali molto veloci. Principali caratteristiche possono essere elencate di seguito:

- Fast Sample Time < 300 ns;
- Hold Time > 100 us;
- Low Input Offset;
- Buffered Output;

In **figura 3** riportiamo una possibile applicazione finalizzata alla conversione A/D. La **figura 4**, invece, descrive un possibile modello delle impedenze di ingresso per la stima dei parametri temporali; con Cs è indicato il condensatore di campionamento che ha un valore di 5 pF. ❏

CODICE MIP 2828860

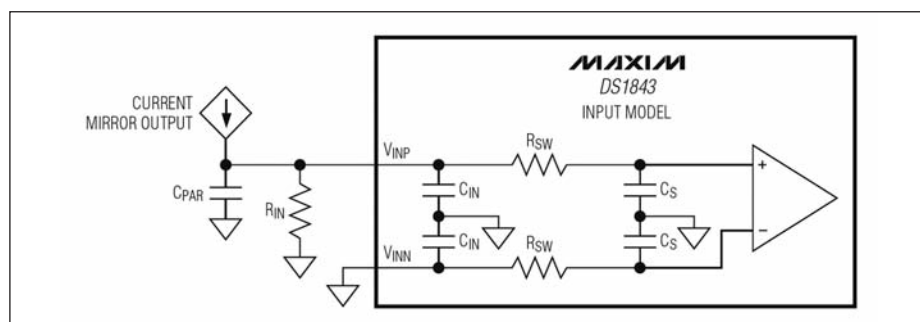



Figura 4: modello del DS1843 per il calcolo di parametri temporali.

di IGINIO COMMISSO (I2UIC)

SEMPLICE *ricevitore* FM

Come realizzare un ricevitore a supereazione per le bande AM e FM utilizzando solo un paio di transistor e pochi altri componenti, alimentato da una batteria da 9 volt.

Nel numero di luglio/agosto di Fare Elettronica abbiamo presentato un ricevitore a onde corte in cui era previsto il dispositivo FONO, che ben si presta ad amplificare il suono anche di questo nuovo ricevitore ciò non toglie che qualsiasi altro amplificatore di BF possa ugualmente andare bene. Utilizzeremo solo due transistor, di cui un FET, precisamente il J310 che può arrivare a gestire anche la banda UHF e si trova facilmente in commercio, anche su Internet. Il secondo transistor è un comune amplificatore di BF, il BC237, facilmente sostituibile da diversi transistor simili, come il BC547. Il circuito è chiamato a supereazione, e si presta molto bene alla demodulazione AM e FM sulle onde dalla VHF in su. Unica avvertenza, non aspettatevi un suono ad alta fedeltà!
Per la sintonia, si è pensato di non utilizzare i classici condensatori variabili a lamelle, perché sono diventati piuttosto costosi e di difficile reperibilità. Abbiamo quindi utilizzato un comune diodo varicap BB329 che ha reso più facile anche il montaggio meccanico.
Il tutto viene facilmente gestito da un potenziometro lineare da 100 kohm.

SCHEMA ELETTRICO

Analizzando lo schema di **figura 1** notiamo che assieme ai transistor citati, si trovano pochi altri componenti, i principale sono la bobina L1/2 e la bobina L3, per il resto si tratta solo di una manciata di resistenze e condensatori.
Il transistor Q1 per rivelare le trasmittenti della banda FM utilizza un circuito, come si è anticipato, a supereazione, che consiste nel portare il transistor in oscillazione e interromperla con un circuito oscillante in BF, permettendo in questo intervallo la rivelazione del suono (R1-C3). Questo circuito consuma pochissimo e può essere facilmente alimentato da una comune pila da 9 volt.
La bobina L3 ha la funzione di separare la parte a radiofrequenza dalla parte di bassa frequenza.
Nella bobina L1/2 ho previsto anche un avvolgimento d'antenna (L2), da utilizzarsi in alternativa alla presa su C9 e trovare la miglior ricezione.
Dalle prove da me fatte (qui a Milano) un filo di mezzo metro, permette già un buon ascolto.

MONTAGGIO

Per il montaggio, questa volta abbiamo utilizzato una basetta quadrata mille fori avente come lati 17 fori (45X45 mm) e il montaggio di **figura 2**.
Per i collegamenti fra i vari componenti si usano i reofori degli stessi, facendo attenzione a non creare cortocircuiti.
Si posizionano per primi i vari pin dell'alimentazione, delle antenne, dell'uscita BF e del potenziometro.
La bobina L1/2 è stata avvolta su un supporto da 5 mm, completo di nucleo, ed è

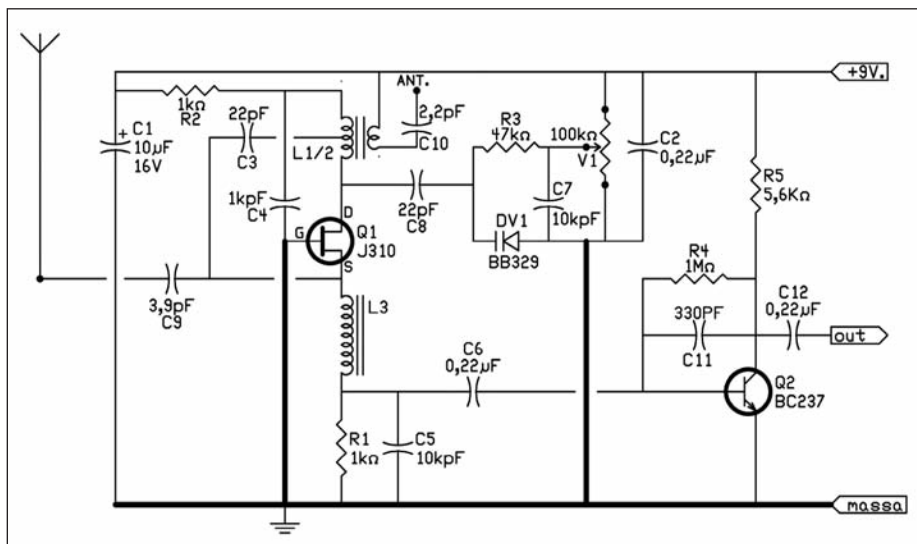



Figura 1: schema elettrico.

composta per L1 da quattro spire di filo smaltato da 0,8 mm con presa al centro. Per L2 bastano due spire dello stesso filo avvolte nella parte superiore della stessa. Non resta che costruire L3, composta da 13 spire, sempre con filo smaltato da 0,8 mm avvolte su un nucleo filettato da 6 mm e utilizzando comodamente i filetti come separatori.
Ora non resta che assemblare il tutto nel migliore dei modi. Come si vede in **figura 2**, bisogna cercare di tenere le piste a RF il più corte possibile onde evitare fenomeni negativi.
La **figura 3** mostra come si presenta la basetta a montaggio terminato.

TARATURA E MESSA A PUNTO

Una volta finito il montaggio non resta che dare corrente e collegare l'amplificatore di BF in ascolto. Sarà bene partire senza il nucleo nella bobina perché, se questo è inserito completamente, blocca l'oscillazione. Si dovrebbero già sentire alcuni rumori che variano al variare della sintonia. Con l'antenna collegata è subito udibile qualche emittente, in questo caso si può usare il nucleo di L1/2 per ridurre l'oscillazione e rendere più udibile il segnale, basta inserirlo per una parte nel supporto. Si potrebbe anche sperimentare quale dei due attacchi d'antenna funziona meglio. Così come previsto, la sintonia dovrebbe essere centrata nella banda FM, comunque la si può ritoccare abbassandola, stringendo le spire di L1 o viceversa alzandola, allargando l'avvolgimento. Nient'altro va toccato. La sintonia va fatta con V1, ruotando lentamente il potenziometro. Va sottolineato il fatto che il potenziometro può essere facilmente posizionato in qualsiasi posizione, cosa non possibile se ci fosse stato un condensatore variabile. Non resta che augurarvi un buon ascolto. ❑

CODICE MIP 2828864

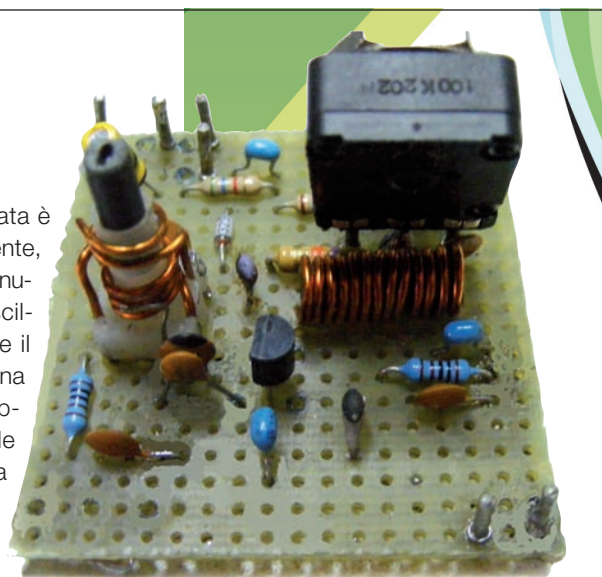



Figura 3: il ricevitore a montaggio terminato.

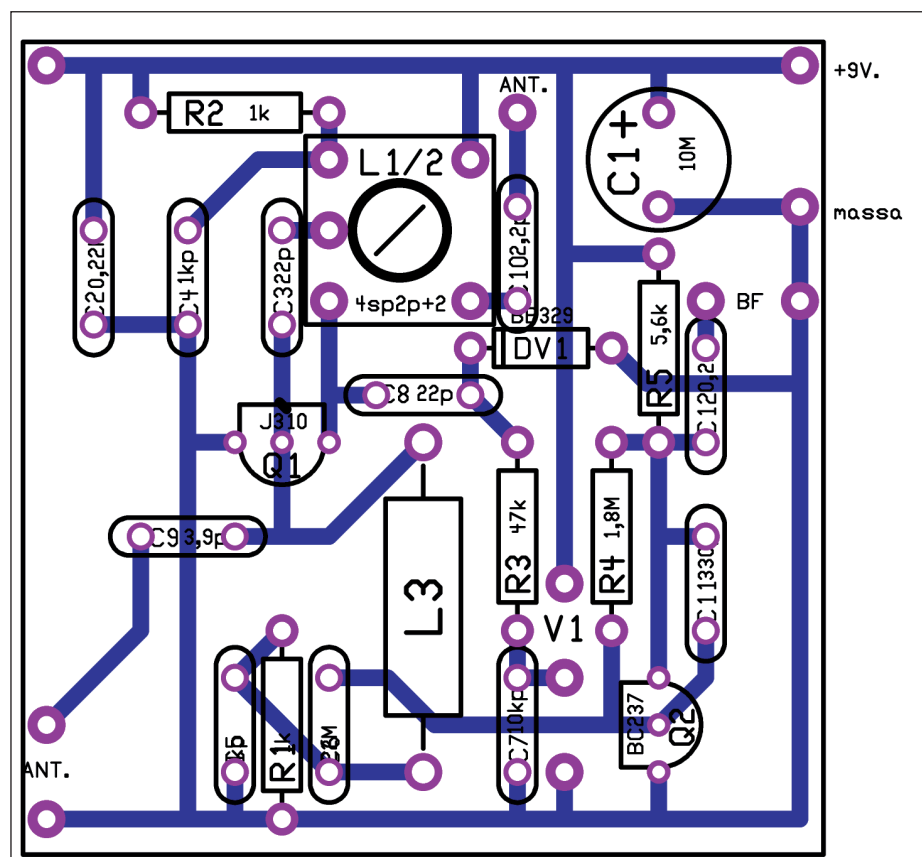



Figura 2: disposizione componenti.

LISTA COMPONENTI

R1-2 = resistenze ¼ watt,1 kohm
R3 = resistenza ¼ watt, 47 kohm
R4 = resistenza ¼ watt, 1,8 Mohm
R5 = resistenza ¼ watt, 5,6 kohm
V1 = potenziometro lineare 100 kohm
C1 = condensatore elettrolitico da 10 MF, 16 Vl.
C2-6-12 = condensatore multistrato 0,22 MF
C3 = condensatore ceramico 22 pF
C4 = condensatore ceramico 1 kpF
C5-7 = condensatore ceramico 10 kpF
C8 = condensatore ceramico 22 pF
C9 = condensatore ceramico 3,9 pF
C10 = condensatore ceramico 2,2 pF
C11 = condensatore ceramico 330 pF
L1/2 = (vedi testo)
L3 = (vedi testo)
Q1 v = transistor FET J310
Q2 = transistor BC237 o equivalente
DV = diodo varicap BB329
1 basetta millefori

LA NUOVA FIERA-LABORATORIO DEDICATA ALL'INVENTIVA E ALL'INTRAPRENDENZA DEI MAKERS E DEI CREATIVI TECNOLOGICI ITALIANI

facebook
twitter
FIERA MILANO

Con il patrocinio di:

- 3D PRINTING & IMAGING
- APP & DIGITAL
- PIATTAFORME OPEN SOURCE
- COMPUTER HARDWARE & SOFTWARE
- CHIMICA
- DESIGN
- ELETTRONICA ED ELETTROTECNICA
- IDROGENO & CELLE A COMBUSTIBILE
- INGEGNERIA
- LEGO AREA
- MATERIE PLASTICHE
- MECCANICA & MECCATRONICA
- MOBILITÀ SOSTENIBILE & VEICOLI ELETTRICI
- RADIO & HAM RADIO
- ROBOTICA
- SOSTENIBILITÀ & ENERGIE ALTERNATIVE
- VIDEO, AUDIO & MOVIE
- WIRELESS ELECTRICITY / TESLA

CODICE MIP 2826456

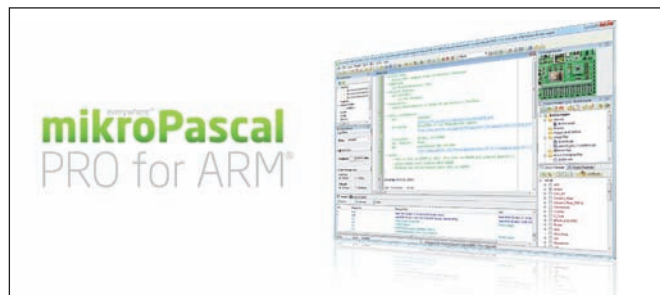
www.makersitaly.it

Lo shop dei prodotti Inware Edizioni...

Nuova piattaforma!

Compilatore Pascal per ARM

mikroPascal PRO for ARM è un compilatore Pascal completo per dispositivi ARMCortex-M3 e Cortex-M4F. Il Pascal è un linguaggio di programmazione popolare con un'ottima sintassi e ha probabilmente il miglior compromesso tra semplicità e controllo. Possiede un IDE intuitivo con supporto docking ricca di caratteristiche, editor di testo avanzato, molti strumenti disponibili, librerie ed esempi. Il compilatore utilizza algoritmi avanzati di ottimizzazione SSA che riducono il codice generato mediamente del 20%. È indubbiamente una soluzione a portata di mano e una scelta eccellente. MikroPascal supporta oltre 450 microcontrollori ARM Cortex-M3 e Cortex-M4 e 186 MCU STM32!

CODICE: MIKROPASCAL-ARM PREZZO: € 252.89

PRIMO PIANO

Arduino Uno SMD Rev3

Arduino Uno è una scheda basata sul microcontrollore Atmel ATmega328 (SMD). Ha 14 Ingressi/uscite digitali (6 delle quali con PWM), 6 ingressi analogici, un cristallo a 16 MHz, una porta USB, una presa di alimentazione, una connessione ICSP, e un bottone di reset. Dispone di tutto il necessario per gestire il microcontrollore a bordo; per utilizzarla basta connettere la scheda al computer con un cavo USB o alimentarla con un alimentatore (max 12 V) o delle batterie esterne. La UNO differisce dalle schede precedenti perché non usa l'FTDI USB-to-serial chip. Le caratteristiche introdotte nella versione R3 sono: ATmega16U2 invece del 8U2 come convertitore USB-seriale. Piedinatura 1.0: sono stati aggiunti i pin SDA e SCL per la comunicazione TWI vicino al pin AREF e vicino al pin RESET altri sono stati collocati altri due nuovi pin, il pin IOREF che consentirà agli shield di adattarsi alla tensione fornita dalla scheda, ed un pin non collegato, riservato per scopi futuri. Circuito di RESET più forte.

Specifiche Tecniche

Microcontrollore: ATmega328 (SMD)
Tensione Operativa di funzionamento: 5 V
Input Voltage (raccomandata): 7-12 V
Tensione di alimentazione massima (non raccomandata): 6-20 V
Pin di I/O Digitali: 14 (6 dei quali forniscono in output segnali PWM)
Pin Input Analogici: 6
Corrente massima ai pin di I/O: 40 mA
Corrente massima per i pin a 3,3 V: 50 mA
Flash Memory: 32 KB (dei quali 0,5 KB usati come bootloader)
SRAM: 2 KB
EEPROM: 1 KB
Frequenza di clock:16 MHz

CODICE: A000073 PREZZO: € 22,99

Shield Wi-Fi per Arduino

Arduino Wi-Fi Shield connette la tua scheda Arduino a Internet in modalità senza fili. Seguendo poche semplici istruzioni è possibile iniziare a controllare tutto il tuo mondo attraverso Internet. Come sempre, con Arduino tutti gli elementi della piattaforma, hardware, software e documenti, sono open-source, scaricabili gratuitamente. Questo significa che è possibile imparare esattamente come è fatta, e utilizzare il progetto come punto di partenza per sviluppare i propri circuiti.

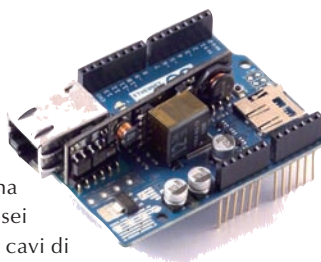
NOVITÀ

Specifiche Tecniche

Richiede la scheda Arduino (non inclusa)
Tensione di alimentazione 5 V (fornita tramite la scheda Arduino)
Connessione tramite rete: 802.11b/g
Modalità Encryption: WEP e WPA2 Personal
Connessione con Arduino tramite porta SPI
on-board micro SD slot
connettori ICSP
connessione FTDI per il serial debugging con il WiFi shield
Micro-USB per l'aggiornamento del firmware WiFi shield
La Wi-Fi Shield comunica con Arduino usando il bus SPI (attraverso il connettore ICSP), quindi è compatibile con tutte le schede che hanno questo tipo di bus.
Questa versione di Wi-Fi shield ha un'antenna integrata al PCB, e basta semplicemente connettere lo shield ad Arduino.

CODICE: A000058 PREZZO: € 83,49

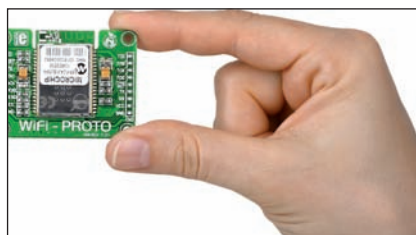
Arduino Ethernet con modulo PoE

Arduino Ethernet è una scheda basato su Arduino UNO, con a bordo un WizNet W5100 TCP/IP Embedded Ethernet Controller. Può essere programmata come una UNO con un connettore seriale tipo FTDI (a sei piedini). L'adattatore USB 2 Serial Arduino o cavi di programmazione FTDI-compatibili possono essere usati per programmarlo. Un modulo PoE (Power Over Ethernet) è saldato sulla scheda per ottenere l'alimentazione da un Cavo Ethernet Categoria 5. È compatibile IEEE802.3af ed è utilizzabile/compatibile con tutti i connettori PoE in commercio.

NOVITÀ

Specifiche Tecniche
Microcontroller: ATmega328
Operating Voltage: 5 V
Input Voltage (recommended): 7-12 V
Input Voltage (limits): 6-20 V
Digital I/O Pins: 14 (of which 4 provide PWM output)
Arduino Pins reserved
- 10 to 13 used for SPI
- 4 used for SD card
- 2 W5100 interrupt (when bridged)
Analog Input Pins: 6
DC Current per I/O Pin: 40 mA
DC Current for 3.3V Pin: 50 mA
Flash Memory: 32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM: 2 KB (ATmega328)
EEPROM: 1 KB (ATmega328)
Clock Speed: 16 MHz
W5100 TCP/IP Embedded Ethernet Controller
Power Over Ethernet ready Magnetic Jack
Micro SD card, with active voltage translators

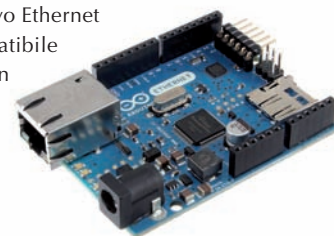
CODICE: A000051 PREZZO: € 61,59

WiFi - PROTO Board

La WiFi - PROTO Board è un modulo wireless 2.4GHz 802.11b miniaturizzato ed economico, ideale per l'uso in applicazioni embedded ed equipaggiamento OEM. Il modulo on-board ZeroG ZG2100M fornisce trasferimento dati estremamente veloce e ampi. I protocolli di comunicazione che possono interfacciare il vostro sistema di controllo o microcontrollore sono comunicazioni standard UART e SPI.

OFFERTA

CODICE: EP-WIFI PREZZO: ~~€ 49,61~~ € 42,35

Arduino Ethernet Rev3

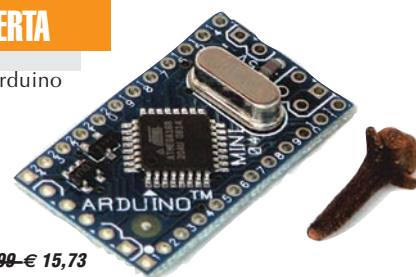
Arduino Ethernet è una scheda a microcontrollore basata sulla Arduino Uno e avente un Controller Ethernet W5100 WIZnet TCP / IP integrato. Può essere programmata come una Uno attraverso un convertitore seriale a sei pin tipo FTDI. Si può usare l'adattatore Arduino USB 2 Serial o qualsiasi cavo tipo FTDI che faccia da convertitore da USB a seriale.
Le caratteristiche introdotte nella versione R3 sono: piedinatura 1.0: sono stati aggiunti i pin SDA e SCL per la comunicazione TWI vicino al pin AREF e vicino al pin RESET altri sono stati collocati altri due nuovi pin, il pin IOREF che consentirà agli shield di adattarsi alla tensione fornita dalla scheda, ed un pin non collegato, riservato per scopi futuri. Circuito di RESET più forte.
Un modulo power-over-Ethernet (PoE), venduto sepratamente, può essere saldato alla scheda per alimentare lo shield da un convenzionale cavo Ethernet intrecciato categoria 5. È compatibile con IEEE802.3af, e funziona con tutti gli iniettori PoE conformi attualmente disponibili.

Specifiche Tecniche
Microcontrollore: ATmega328
Tensione di lavoro: 5 V
Tensione d'Ingresso (raccomandata): 7-12 V
Tensione d'Ingresso (limiti): 6-20 V
I/O Digitali: 14 (di cui 4 usati anche per le uscite PWM)
Pin di Arduino riservati
da 10 a 13 usati per SPI
4 usato per la scheda SD
2 usato dal W5100
Ingressi Analogici: 6
Corrente DC per ogni pin I/O: 40 mA
Corrente DC dal pin 3,3 V: 50 mA
Memoria Flash: 32 KB (ATmega328) di cui 0.5 KB usati dal bootloader
SRAM: 2 KB (ATmega328)
EEPROM: 1 KB (ATmega328)
Velocità clock: 16 MHz
W5100 TCP/IP Controller Ethernet incorporato
Power Over Ethernet ready Magnetic Jack
Micro SD slot, con convertitore di livelli attivo

NOVITÀ

CODICE: A000068 PREZZO: € 48,28

Arduino Mini Light

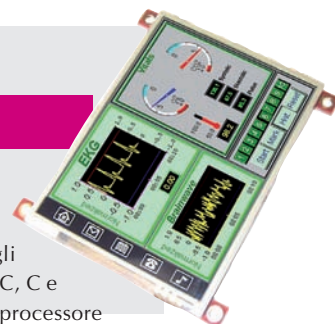
OFFERTA

Arduino Mini Light è la versione micro miniaturizzata della scheda Arduino (senza parte USB) sprovvista di connettore downside. Le dimensioni estremamente ridotte (30x18 mm) permettono un diminuzione dello spazio di ingombro non indifferente. Fa uso di ATMega168 con 16K di spazio programma ed è utilizzabile con convertitore USB/Seriale per la programmazione e l'aggiunta della porta USB.

CODICE: A000004 PREZZO: ~~€ 22,99~~ € 15,73

Lo shop dei prodotti Inware Edizioni... ma non solo!

BEST SELLER

Modulo display da 3.2"

uLCD-32PT(GFX) è un modulo LCD "SMART" da 3,2" all-in-one compatto ed economico che fa uso della più recente tecnologia LCD a matrice attiva (TFT) e dispone di un controller grafico PICASO-GFX2 integrato che fornisce funzionalità "stand-alone" a qualsiasi progetto. Il PICASO-GFX2 appartiene a una famiglia di processori che dispone di un motore virtuale soft core altamente ottimizzato, E.V.E. (Extensible Virtual Engine). EVE è un processore virtuale proprietario ad alte prestazioni con un set di istruzioni byte-code estensivo ottimizzato per eseguire programmi compilati in 4DGL. 4DGL (4D Graphics Language) fu specificatamente sviluppato da zero per l'engine core EVE. È un linguaggio ad alto livello di facile apprendimento e semplice da capire, eppure abbastanza potente per affrontare tantissime applicazioni grafiche embedded. 4DGL permette allo sviluppatore di scrivere applicazioni a un alto livello di sintassi simile per gli altri linguaggi più popolari quali BASIC, C e Pascal, eseguendole direttamente nel processore PICASO-GFX2 integrato nel modulo uLCD-32PT. Permette all'utente di assumere il pieno controllo di tutte le risorse disponibili sulla piattaforma hardware quali porte seriali, display LCD grafico, memory card uSD, pins I/O ecc. Questo elimina il bisogno di avere un processore/controller esterno per pilotare il modulo uLCD-32PT via comandi seriali. Fornisce all'utente il completo controllo sul modulo hardware permettendogli di sviluppare potenti applicazioni.

CODICE: ULCD-32-PT PREZZO: € 79,86

Annate Firmware 2006/2011

Tutti i numeri di Firmware in un comodo file zip da scaricare! In pdf ad alta risoluzione tutti i numeri della rivista dal 2006 al 2011.

CODICE: FW-DVD PREZZO: € 20,16

MOACON

MOACON è un controllore industriale modulare programmabile in C. Grazie al suo design modulare, il MOACON consente ai progettisti di adottare solo i moduli necessari per i requisiti specifici di progetto, permettendo una ottimale adattabilità alle esigenze specifiche.

Caratteristiche di base del MOACON

1. Design modulare
2. Programmabile in linguaggio C
3. Basato su processore ARM a 32 bit
4. Download e debug tramite porta USB
5. MOACON Studio - Ambiente di sviluppo gratuito (IDE)

Il MOACON, grazie alla CPU ARM 32-bit permette elevate prestazioni. Sono disponibili singoli moduli con funzionalità di I/O digitali, uscita a relè, conversione analogico-digitale e digitale-analogico, controllo motore, controllo della temperatura, RS-232 e Ethernet, e potenzialmente molti altri.

L'ambiente di sviluppo integrato e gratuito, il MOACON Studio, dispone di un compilatore C, editor, RS-232, il download e il debug tramite USB, e altre funzionalità. L'editor permette l'evidenziazione della sintassi, il completamento automatico dei comandi, help sensibile al contesto tutto per rendere l'apprendimento, lo sviluppo, il test dei progetti con MOACON ad elevata produttività oltre ad essere una piacevole esperienza.

CODICE: DP-CPU500 PREZZO: € 133,10

Sviluppare con Texas

ASLK PRO è stato prodotto da mikroElektronika per Texas Instruments. Questo kit è progettato per fornire a studenti e tecnici una piattaforma per la creazione di sistemi analogici. L'idea principale dietro questo kit è di fornire una piattaforma economica per realizzare quasi qualsiasi sistema analogico utilizzando circuiti integrati general-purpose quali amplificatori operazionali, moltiplicatori analogici. ASLK PRO è fornito di tre amplificatori operazioni general-purpose (TL082) e tre moltiplicatori analogici a larga banda MPY634 di Texas Instruments. Sono inclusi, inoltre, due convertitori-moltiplicatori da digitale ad analogico a 12 bit input-parallelo DAC7821, e un controller DC/DC di tipo buck TPS40200 con ampio ingresso non sincorono e un regolatore a basso droput TPS7250 sempre di Texas Instruments.
Una parte della scheda di ASLK PRO è lasciata per la prototipazione generica che può essere usata per la realizzazione di mini progetti.
La scheda è corredata da un manuale user-friendly che guiderà l'utente attraverso 14 esercitazioni di laboratorio che aiuteranno a capire i moduli presenti sulla scheda attraverso esperimenti molto interessanti e utili. Questi esempi sono un buon punto di partenza per tutti i progetti futuri. Il manuale incluso è di facile lettura, con buone spiegazioni e tantissime foto esplicative.

CODICE: ANSYS-LK-PRO PREZZO: € 168,19

prossimamente su Fare Elettronica

COMING...

RASPBERRY PI

Applicazioni con Raspberry PI.
Sfruttiamo al massimo le potenzialità del Raspberry PI per costruire numerose applicazioni: dai videogiochi alla cyclette intelligente!

TUTORIAL PIC18

Se avete seguito le prime lezioni, uscite sui precedenti numeri di Fare Elettronica, ora avete le nozioni necessarie per interfacciare al PIC molti tipi di sensori e attuatori. Utilizzando questo tipo di periferiche si ha spesso la necessità di visualizzare le informazioni elaborate dal PIC in una forma chiara e leggibile.

SENSORI CAPACITIVI

Grazie all'estrema versatilità e robustezza, la trasduzione capacitiva è alla base di numerosi sensori industriali (ad esempio di distanza, di livello e tomografici), di molti nuovi sensori MEMS e delle moderne interfacce uomo-macchina, come touch-screen, pulsanti e cursori a sfioramento. Analizziamo quindi i principi di trasduzione, le diverse configurazioni, gli integrati di misura e le principali applicazioni.

COME RISPARMIARE?

Ordinare direttamente tramite internet conviene! Potrai infatti aggiudicarti dei buoni sconto (coupon) con le modalità:

COUPON ABBONAMENTO ONLINE FE E FW
Sottoscrivendo un nuovo abbonamento o rinnovando il proprio online, direttamente tramite IE SHOP, si ha diritto ad un coupon del 15%! **Qualora ci si abboni (sempre online) a Fare Elettronica e Firmware insieme, il coupon è del 20%!**
COUPON QUIZ "ELETTROQUIZ"
Rispondendo ai Quiz presente ogni mese su Fare Elettronica si avrà diritto a un coupon del 10%. Tutte le risposte pervenute saranno accuratamente valutate dagli autori dei quesiti e, in caso di risposta esatta, (oltre al Coupon) potrai vincere fantastici premi!
COUPON "ACQUISTI PREMIATI"
I tuoi acquisti su www.elettroshop.com/ieshop vengono premiati con un coupon del 10% se il tuo ordine supera i 100 EUR (IVA e spese di trasporto escluse). Ma se il tuo ordine supera i 200 EUR, lo sconto sale al 20%!
IMPORTANTE!

- Il coupon può essere speso **esclusivamente** ordinando sul sito www.elettroshop.com/ieshop
- Il coupon non potrà essere utilizzato in caso di nuova sottoscrizione o rinnovo dell'abbonamento, è invece utilizzabile per tutti gli altri prodotti presenti su www.elettroshop.com/ieshop
- Ogni coupon ha una scadenza, non dimenticarlo!
- Lo slogan "più acquisti più risparmi" pecca senz'altro di originalità, ma in questo caso è molto azzeccato: i coupon sono infatti utilizzabili solo una volta, usali bene!
- I coupon non sono cumulabili, e vanno utilizzati singolarmente, secondo il criterio cronologico di assegnazione.

CONTI CORRENTI POSTALI - Ricevuta di Accredito BancoPosta

€ sul C/C n. 70107552 di Euro

importo in lettere

TD 451

INTESTATO A:
INWARE EDIZIONI SRL

CAUSALE

ESEGUITO DA:
VIA - PIAZZA
CAP
LOCALITÀ

BOLLO DELL'UFF. POSTALE
codice bancoposta

IMPORTANTE: NON SCRIVERE NELLA ZONA SOTTOSTANTE
importo in euro numero conto td

70107552 < 451>

AUT. DB/SISB/E 20017 DEL 11.04.2006

CONTI CORRENTI POSTALI - Ricevuta di Versamento BancoPosta

€ sul C/C n. 70107552 di Euro

importo in lettere

INTESTATO A:
INWARE EDIZIONI SRL

ESEGUITO DA:
VIA - PIAZZA
CAP
LOCALITÀ

BOLLO DELL'UFF. POSTALE

AVVERTENZE
Il Bollettino deve essere compilato in ogni sua parte (con inchiostro nero o blu) e non deve recare abrasioni, correzioni o cancellature.
La causale è obbligatoria per i versamenti a favore delle Pubbliche Amministrazioni.
Le informazioni richieste vanno riportate in modo identico in ciascuna delle parti di cui si compone il bollettino.

CAUSALE

IMPORTANTE: NON SCRIVERE NELLA ZONA SOPRASTANTE!

AVVERTENZE

Il Bollettino deve essere compilato in ogni sua parte (con inchiostro nero o blu) e non deve recare abrasioni, correzioni o cancellature.

La causale è obbligatoria per i versamenti a favore delle Pubbliche Amministrazioni.

Le informazioni richieste vanno riportate in modo identico in ciascuna delle parti di cui si compone il bollettino.

Indagine sui lettori

Aiutaci a conoscerti meglio!

Con il tuo aiuto riusciremo ad offrirti una rivista sempre più in linea con le tue aspettative.

Compila online il questionario all'indirizzo www.farelettronica.com/survey

Per ringraziarti per il tuo tempo e la tua cortesia, ti invieremo gratuitamente un bellissimo eBook del valore di 14,52 euro!

PRODOTTI

Se vuoi ordinare "offline" ovvero senza usare Internet, usa il seguente modulo d'ordine **PRODOTTI**

COD. PRODOTTO	DESCRIZIONE	PREZZO UNITARIO	Q.tà	Totale
			SUB TOTALE	
			SPESE DI SPEDIZIONE	
			TOTALE	

Nota. Se le righe non sono sufficienti, utilizzare più copie del modulo.
Per maggiori dettagli sulle spese di spedizione e i metodi di pagamento vedi retro cartolina.

METODI DI PAGAMENTO

☐ **BONIFICO BANCARIO** ☐ **BOLLETTINO POSTALE** ☐ **CONTRASSEGNO**

☐ MasterCard ☐ VISA ☐ postepay

TITOLARE. . . . .
NUMERO . . . .
DATA DI SCADENZA. . . . . CODICE DI SICUREZZA . . . .

FE 328

INVIARE IL MODULO D'ORDINE A: Inware Edizioni srl, Via Giotto, 7 - 20032 Cormano (MI) - oppure via FAX allo 02 66508225
(Inviando questo modulo si accettano i termini e le condizioni riportate a tergo)

Nome . . . . Cognome . . . .
Indirizzo . . . . n° . . . .
Cap . . . . Città . . . . Prov . . . .
Tel . . . . Fax . . . . Email . . . .
Ragione Sociale . . . .
Codice fiscale . . . . Partita IVA . . . .

DATA . . . . FIRMA . . . . ☐ Fattura

ABBONAMENTO

Sì, mi abbono a Fare Elettronica

☐ Abbonamento TRIAL a 3 numeri di FARE ELETTRONICA a soli € 16,50 anziché € 18,00
☐ Abbonamento PREMIUM a 11 numeri di FARE ELETTRONICA a soli € 55,00 anziché € 66,00
☐ Abbonamento PRO a 11 numeri di FARE ELETTRONICA include il CD dell'annata 2011 a soli € 59,50 anzichè € 96,00

FE 328

METODI DI PAGAMENTO (per maggiori dettagli vedi retro cartolina)

☐ **BONIFICO BANCARIO** ☐ **BOLLETTINO POSTALE** ☐ **ALLEGO ASSEGNO** (Intestato a Inware Edizioni)

☐ MasterCard ☐ VISA ☐ postepay

PayPal fare il pagamento su paypal@inwaredizioni.it

TITOLARE. . . . .
NUMERO . . . .
DATA DI SCADENZA. . . . . CODICE DI SICUREZZA . . . .

INVIARE IL MODULO D'ORDINE A: Inware Edizioni srl, Via Giotto, 7 - 20032 Cormano (MI) - oppure via FAX allo 02 66508225
(Inviando questo modulo si accettano i termini e le condizioni riportate a tergo)

Nome . . . . Cognome . . . .
Indirizzo . . . . n° . . . .
Cap . . . . Città . . . . Prov . . . .
Tel . . . . Fax . . . . Email . . . .
Ragione Sociale . . . .
Codice fiscale . . . . Partita IVA . . . .

DATA . . . . FIRMA . . . . ☐ Fattura

ISTRUZIONI PER L'ORDINE E SPESE DI SPEDIZIONE

Il modulo d'ordine dovrà essere compilato in tutte le sue parti ed inviato via posta o via fax ai recapiti indicati sul modulo d'ordine stesso. Gli ordini potranno essere fatti anche direttamente online dal sito www.elettroshop.com/ieshop. In questo caso non sarà necessario inviare il modulo d'ordine. Le spese di spedizione ammontano a euro 8,50 a cui vanno aggiunti euro 3,50 se si sceglie di pagare in contrassegno.

METODI DI PAGAMENTO

Si accettano pagamenti ino in contrassegno, carta di credit, bollettino postale o bonifico all'ordine. Per il contrassegno verrà applicata una spesa aggiuntiva di euro 3,50 per le spese di contrassegno. Forme diverse di pagamento devono essere previamente concordate.

Bonifico bancario

Appoggiato su Poste Italiane IBAN: IT 68 I 07601 01600 000070107552 intestato a Inware Edizioni srl.

Bollettino postale

Versamento sul conto corrente postale n. 70107552 intestato a Inware Edizioni srl. Inviare la ricevuta (o copia) del versamento.

Contrassegno

La merce verrà pagata direttamente al corriere alla consegna della merce. Il pagamento in contrassegno comporta l'addebito di euro 3,50 per spese di contrassegno.

Carta di Credito

Il pagamento con carta di credito può essere effettuato specificando i dati della carta via posta, email, web, fax o telefono. Per gli ordini effettuati sul sito il pagamento mediante carta di credito verrà effettuato attraverso una connessione sicura SSL per garantire la massima sicurezza.

TERMINI E CONDIZIONI

Tutti i prodotti sono coperti da garanzia. La garanzia copre tutti i difetti di fabbricazione entro un anno dal ricevimento della merce. Tutti i prodotti non funzionanti per uso improprio o incuria non saranno ritenuti in garanzia e saranno addebitati gli eventuali costi di riparazione. Tutti i prodotti verranno riparati e/o sostituiti direttamente dal produttore. Non sono coperti da garanzia i componenti elettronici (microprocessori, memorie, ecc.) La garanzia dei prodotti si intende F.co ns. sede, le eventuali spese di trasporto sono a carico del cliente salvo accordi diversi. Per dar corso alla riparazione/sostituzione in garanzia è necessario seguire l'apposita procedura di RMA.

PRIVACY

Ai sensi del Decr. Lgs. 196/2003 la informiamo che i dati trasmessi verranno impiegati coi principali scopi di indagini di mercato e nelle modalità previste dallo stesso, prevalentemente con mezzi informatici. Il conferimento, di norma facoltativo, è obbligatorio per permettere il rapporto commerciale. È in ogni caso fatto diritto dell'interessato esercitare i propri diritti, nei modi previsti dal "Titolo II art. 7" della legge sopra citata, scrivendo a Inware Edizioni srl, Via Giotto 7 – 20032 Cormano o tramite email all'indirizzo info@inwaredizioni.it

---

ABBONARSI ALLE RIVISTE INWARE EDIZIONI CONVIENE!

I vantaggi per gli abbonati sono, oltre al prezzo bloccato per un anno, la ricezione del numero direttamente a casa con la garanzia di ricevere tutti i numeri. Inoltre un vistoso risparmio che, nel caso dell'abbonamento PLUS, ammonta a ben 36,50 euro. L'abbonamento ha una durata di 12 mesi e comporta l'invio di 11 numeri di Fare Elettronica. Eventuali variazioni di indirizzo andranno comunicate tempestivamente alla Redazione che provvederà a registrare il cambiamento senza alcuna spesa aggiuntiva. L'abbonamento decorrerà dal primo numero raggiungibile alla data di avvenuto pagamento. Non sono previsti rimborsi in caso di disdetta dell'abbonamento.

METODI DI PAGAMENTO

Bonifico bancario

Appoggiato su Poste Italiane IBAN: IT 68 I 07601 01600 000070107552 intestato a Inware Edizioni srl.

Bollettino postale

Versamento sul conto corrente postale n. 70107552 intestato a Inware Edizioni srl. Inviare la ricevuta (o copia) del versamento indicando nella causale:
"Abbonamento Fare Elettronica"

Carta di Credito

Il pagamento con carta di credito può essere effettuato specificando i dati della carta via posta, email, web, fax o telefono. Per gli ordini effettuati sul sito il pagamento mediante carta di credito verrà effettuato attraverso una connessione sicura SSL per garantire la massima sicurezza.

Assegno bancario

È possibile spedire un assegno bancario insieme a questo coupon.

Paypal

Inviare il pagamento a paypal@invaredizioni.it.

NUOVA PIATTAFORMA!

elettroshop.com

brilliant electronics since 1998

Spedizione omaggio per un ordine a partire da soli 30 EUR ai primi 20 clienti! Codice coupon: 59FC1A09CB39FB1

www.elettroshop.com